



---

**CYBER 96X MSL  
MAINTENANCE SOFTWARE  
REFERENCE MANUAL**

**LMT3/P TEST DESCRIPTION**

---

**CDC® COMPUTER SYSTEMS  
CYBER 960, 962**

# PREFACE

-----

This manual describes the CDC CYBER 960 and 962 Computer Systems LMT3/P test. Procedures for using this test are provided in the CYBER 96X Test Procedures Reference Manual.

## NOTE

It is assumed and planned that the user of program descriptions will have an in-depth understanding of the processor logic; including familiarity with all levels of the Multi-Level Block Diagrams.

## DISCLAIMER

This product is intended for use only as described in this document. Control Data cannot be responsible for the proper functioning of undescribed features or undefined parameters.

80000285 A

LMT3/P - 1

| REVISION RECORD |                                              |
|-----------------|----------------------------------------------|
| REVISION        | DESCRIPTION                                  |
| A<br>(07-11-88) | Manual released.                             |
| B<br>(11-08-88) | Manual updated to reflect technical changes. |

Publication No.

\_\_\_80000285\_\_\_

REVISION LETTERS I, O, Q, S, X, AND Z ARE NOT USED

1988

by Control Data Corporation  
Printed in the United States of America

80000285 B

Address comments concerning  
this manual to:

Control Data Corporation  
Technology and Graphics Division  
4201 Lexington Avenue North  
St. Paul, Minnesota 55128-8198

or use Comment Sheet in  
the back of this manual.

LMT3/P - 11



# CONTENTS

|                                                 |     |
|-------------------------------------------------|-----|
| General                                         | 1   |
| Section Descriptions                            | 9   |
| Section 0 - RMA Address Path                    | 9   |
| Section 0, Subsection 0                         | 9   |
| Section 0, Subsection 1                         | 11  |
| Section 0, Subsection 2                         | 14  |
| Section 0, Subsection 3                         | 18  |
| Section 0, Subsection 4                         | 20  |
| Section 1 - Parity Circuits Using RMA Addresses | 23  |
| Section 1, Subsection 0                         | 23  |
| Section 1, Subsection 1                         | 26  |
| Section 1, Subsection 2                         | 29  |
| Section 1, Subsection 3                         | 31  |
| Section 1, Subsection 4                         | 34  |
| Section 1, Subsection 5                         | 36  |
| Section 2 - Preliminary Cache Checks            | 39  |
| Section 2, Subsection 0                         | 39  |
| Section 2, Subsection 1                         | 42  |
| Section 3 - Cache Data RAMs Data Integrity      | 47  |
| Section 3, Subsection 0                         | 47  |
| Section 3, Subsection 1                         | 50  |
| Section 3, Subsection 2                         | 53  |
| Section 3, Subsection 3                         | 58  |
| Section 4 - Cache Tag RAMs Data Integrity       | 60  |
| Section 4, Subsection 0                         | 60  |
| Section 4, Subsection 1                         | 63  |
| Section 4, Subsection 2                         | 66  |
| Section 4, Subsection 3                         | 68  |
| Section 5 - Cache Tag Compare Network           | 72  |
| Section 5, Subsection 0                         | 72  |
| Section 5, Subsection 1                         | 75  |
| Section 5, Subsection 2                         | 78  |
| Section 5, Subsection 3                         | 81  |
| Section 6 - Cache Addressing, External to RAM   | 84  |
| Section 6, Subsection 0                         | 84  |
| Section 6, Subsection 1                         | 87  |
| Section 6, Subsection 2                         | 90  |
| Section 6, Subsection 3                         | 93  |
| Section 7 - Cache RAM Addressing, Tag and Data  | 97  |
| Section 7, Subsection 0                         | 97  |
| Section 7, Subsection 1                         | 100 |
| Section 7, Subsection 2                         | 103 |
| Section 7, Subsection 3                         | 105 |

|                                                  |     |
|--------------------------------------------------|-----|
| Section 8 - Purge Cache All                      | 109 |
| Section 8, Subsection 0                          | 109 |
| Section 8, Subsection 1                          | 112 |
| Section 8, Subsection 2                          | 114 |
| Section 8, Subsection 3                          | 118 |
| Section 9 - Cache Usable Signal from MAC         | 119 |
| Section 9, Subsection 0                          | 119 |
| Section 9, Subsection 1                          | 121 |
| Section 9, Subsection 2                          | 123 |
| Section 9, Subsection 3                          | 125 |
| Section 10 - Cache Parity Circuits               | 128 |
| Section 10, Subsection 0                         | 128 |
| Section 10, Subsection 1                         | 130 |
| Section 10, Subsection 2                         | 133 |
| Section 10, Subsection 3                         | 135 |
| Section 10, Subsection 4                         | 137 |
| Section 11 - Cache Parity Error Signals          | 139 |
| Section 11, Subsection 0                         | 139 |
| Section 11, Subsection 1                         | 141 |
| Section 11, Subsection 2                         | 143 |
| Section 11, Subsection 3                         | 145 |
| Section 11, Subsection 4                         | 149 |
| Section 11, Subsection 5                         | 151 |
| Section 11, Subsection 6                         | 153 |
| Section 12 - Purge Cache Functions               | 156 |
| Section 12, Subsection 0                         | 157 |
| Section 12, Subsection 1                         | 161 |
| Section 12, Subsection 2                         | 166 |
| Section 13 - Cache Allocate and LRU Network      | 170 |
| Section 13, Subsection 0                         | 170 |
| Section 13, Subsection 1                         | 175 |
| Section 13, Subsection 2                         | 180 |
| Section 13, Subsection 3                         | 184 |
| Section 14 - Address Control A Stream Counter    | 187 |
| Section 14, Subsection 0                         | 187 |
| Section 15 - Page Table Address/Length Registers | 191 |
| Section 15, Subsection 0                         | 191 |
| Section 15, Subsection 1                         | 194 |
| Section 16 - Page Table Address to RMA Mux       | 197 |
| Section 16, Subsection 0                         | 197 |
| Section 16, Subsection 1                         | 199 |
| Section 17 - Hash Code Logic Network             | 202 |
| Section 17, Subsection 0                         | 202 |
| Section 17, Subsection 1                         | 208 |
| Section 18 - The RMA Incrementer Test            | 209 |
| Section 18, Subsection 0                         | 209 |
| Section 18, Subsection 1                         | 212 |
| Section 19 - RMA Mux - Incrementer Logic Merge   | 216 |
| Section 19, Subsection 0                         | 216 |
| Section 19, Subsection 1                         | 219 |
| Section 20 - PTA/PTL Parity Circuits             | 223 |
| Section 20, Subsection 0                         | 223 |
| Section 20, Subsection 1                         | 225 |

|                                                                 |     |
|-----------------------------------------------------------------|-----|
| Section 21 - Page Table Hit Comparator .....                    | 228 |
| Section 21, Subsection 0 .....                                  | 228 |
| Section 21, Subsection 1 .....                                  | 231 |
| Section 22 - RMA Mux, Page Frame Address Entry .....            | 236 |
| Section 22, Subsection 0 .....                                  | 237 |
| Section 22, Subsection 1 .....                                  | 239 |
| Section 23 - Map PFA/PO Merge Input to Final Mux Register ..... | 244 |
| Section 23, Subsection 0 .....                                  | 245 |
| Section 23, Subsection 1 .....                                  | 249 |
| Section 23, Subsection 2 .....                                  | 252 |
| Section 24 - Page Map Data Integrity .....                      | 255 |
| Section 24, Subsection 0 .....                                  | 255 |
| Section 24, Subsection 1 .....                                  | 258 |
| Section 24, Subsection 2 .....                                  | 261 |
| Section 24, Subsection 3 .....                                  | 265 |
| Section 25 - Page Map Addressing .....                          | 269 |
| Section 25, Subsection 0 .....                                  | 269 |
| Section 25, Subsection 1 .....                                  | 273 |
| Section 25, Subsection 2 .....                                  | 276 |
| Section 26 - Purge Page Map All .....                           | 280 |
| Section 26, Subsection 0 .....                                  | 280 |
| Section 26, Subsection 1 .....                                  | 283 |
| Section 26, Subsection 2 .....                                  | 286 |
| Section 27 - Complete Map Purge Test .....                      | 289 |
| Section 27, Subsection 0 .....                                  | 289 |
| Section 28 - Page Map Comparator Logic .....                    | 293 |
| Section 28, Subsection 0 .....                                  | 293 |
| Section 28, Subsection 1 .....                                  | 297 |
| Section 28, Subsection 2 .....                                  | 301 |
| Section 29 - Page Map Status RAM and Allocate Test .....        | 305 |
| Section 29, Subsection 0 .....                                  | 305 |
| Section 29, Subsection 1 .....                                  | 310 |
| Section 30 - Page Map Parity Logic .....                        | 314 |
| Section 30, Subsection 0 .....                                  | 314 |
| Section 30, Subsection 1 .....                                  | 317 |
| Section 30, Subsection 2 .....                                  | 321 |
| Section 31 - Page Table Search and Basic Block Fill .....       | 324 |
| Section 31, Subsection 0 .....                                  | 324 |
| Section 31, Subsection 1 .....                                  | 327 |
| Section 32 - CM to Cache Data Path and Tag File Test .....      | 334 |
| Section 32, Subsection 0 .....                                  | 335 |
| Section 32, Subsection 1 .....                                  | 339 |
| Section 32, Subsection 2 .....                                  | 343 |
| Section 32, Subsection 3 .....                                  | 348 |
| Section 32, Subsection 4 .....                                  | 352 |
| Section 32, Subsection 5 .....                                  | 356 |
| Section 32, Subsection 6 .....                                  | 361 |
| Section 32, Subsection 7 .....                                  | 365 |
| Section 33 - Cache Hit Valid Logic .....                        | 370 |
| Section 33, Subsection 0 .....                                  | 370 |
| Section 33, Subsection 1 .....                                  | 373 |

|                                                                             |     |
|-----------------------------------------------------------------------------|-----|
| Section 34 - Cache Hit Not Valid and Associated RAMs .....                  | 378 |
| Section 34, Subsection 0 .....                                              | 379 |
| Section 34, Subsection 1 .....                                              | 384 |
| Section 34, Subsection 2 .....                                              | 389 |
| Section 34, Subsection 3 .....                                              | 394 |
| Section 35 - CMC Tag Decoder Rank A/Rank B Logic and Data .....             | 400 |
| Section 35, Subsection 0 .....                                              | 400 |
| Section 35, Subsection 1 .....                                              | 405 |
| Section 35, Subsection 2 .....                                              | 409 |
| Section 36 - Fakecm and Fault Responses .....                               | 413 |
| Section 36, Subsection 0 .....                                              | 413 |
| Section 36, Subsection 1 .....                                              | 415 |
| Section 36, Subsection 2 .....                                              | 419 |
| Section 36, Subsection 3 .....                                              | 421 |
| Section 36, Subsection 4 .....                                              | 423 |
| Section 37 - CM Invalidate/Cache Bypass .....                               | 424 |
| Section 37, Subsection 0 .....                                              | 424 |
| Section 37, Subsection 1 .....                                              | 429 |
| Section 37, Subsection 2 .....                                              | 434 |
| Section 38 - Tag File, Address Mux, Block Fill Data Parity<br>Circuit ..... | 436 |
| Section 38, Subsection 0 .....                                              | 436 |
| Section 38, Subsection 1 .....                                              | 439 |
| Section 38, Subsection 2 .....                                              | 443 |
| Section 39 - Cache Lookahead, Part 1 .....                                  | 448 |
| Section 39, Subsection 0 .....                                              | 448 |
| Section 39, Subsection 1 .....                                              | 452 |
| Section 39, Subsection 2 .....                                              | 454 |
| Section 40 - Cache Lookahead, Part 2 .....                                  | 459 |
| Section 40, Subsection 0 .....                                              | 460 |
| Section 40, Subsection 1 .....                                              | 463 |
| Section 40, Subsection 2 .....                                              | 466 |
| Section 40, Subsection 3 .....                                              | 469 |
| Section 41 - Cache Lookahead, Part 3 .....                                  | 474 |
| Section 41, Subsection 0 .....                                              | 474 |
| Section 41, Subsection 1 .....                                              | 477 |
| Section 41, Subsection 2 .....                                              | 480 |
| Section 42 - Page Table Update .....                                        | 485 |
| Section 42, Subsection 0 .....                                              | 485 |
| Section 42, Subsection 1 .....                                              | 488 |
| Section 42, Subsection 2 .....                                              | 491 |
| Section 42, Subsection 3 .....                                              | 494 |
| Section 43 - Page Map Status RAM, Modify Bit .....                          | 498 |
| Section 43, Subsection 0 .....                                              | 498 |
| Section 43, Subsection 1 .....                                              | 501 |
| Section 43, Subsection 2 .....                                              | 503 |
| Section 44 - Page Map Status Parity Circuits (MPLRPE) .....                 | 507 |
| Section 44, Subsection 0 .....                                              | 507 |
| Section 44, Subsection 1 .....                                              | 510 |
| Section 45 - LMMIC Function Code Parity Logic .....                         | 513 |
| Section 45, Subsection 0 .....                                              | 513 |
| Section 45, Subsection 1 .....                                              | 515 |

|                                                         |     |
|---------------------------------------------------------|-----|
| Section 46 - Test the CAR Equal (FESE) Comparator ..... | 518 |
| Section 46, Subsection 0 .....                          | 518 |
| Section 46, Subsection 1 .....                          | 523 |
| Section 46, Subsection 2 .....                          | 526 |
| Section 47 - Conflict Testing, Allocation Delays .....  | 533 |
| Section 47, Subsection 0 .....                          | 533 |
| Section 48 - Cache/Page Map Conflicts .....             | 538 |
| Section 48, Subsection 0 .....                          | 538 |
| Section 48, Subsection 1 .....                          | 541 |
| Section 48, Subsection 2 .....                          | 545 |
| Section 48, Subsection 3 .....                          | 548 |
| Section 49 - Invalidate Conflicts .....                 | 551 |
| Section 49, Subsection 0 .....                          | 552 |
| Section 49, Subsection 1 .....                          | 554 |
| Section 49, Subsection 2 .....                          | 557 |
| Section 49, Subsection 3 .....                          | 560 |
| Section 49, Subsection 4 .....                          | 562 |
| Section 49, Subsection 5 .....                          | 564 |
| Section 50 - Test Function Code Bits .....              | 569 |
| Section 50, Subsection 0 .....                          | 569 |
| Section 51 - Test Map Hit Signal .....                  | 571 |
| Section 51, Subsection 0 .....                          | 574 |
| Section 51, Subsection 1 .....                          | 576 |

#### APPENDIXES

|                                     |     |
|-------------------------------------|-----|
| A. GLOSSARY .....                   | A-1 |
| B. LMT3/P MICRAND DEFINITIONS ..... | B-1 |

#### FIGURE

|                            |    |
|----------------------------|----|
| LMT3/P-1 Typical SVA ..... | 60 |
|----------------------------|----|

#### LOCAL MEMORY TEST DESCRIPTION LMT3/P

=====

#### GENERAL

LMT3/P tests the processor Local Memory logic block and provides error detection of all hardware failures. Failure data, when analyzed, aids hardware isolation.

The diagnostic consists of 52 sections. The following defines the major areas tested by these sections.

- o Sections 0 through 1: These test elements complete Real Memory Address (RMA) testing. Several interface lines between LM and CMC as well as some circuitry in CMC are tested.
- o Sections 2 through 13: These sections perform Basic Cache tests. FAKECM mode is the primary aid and is used to test Cache control and Cache set logic.
- o Section 14: This section tests Address Control logic. It completes testing of the A-Stream Address Adder.
- o Sections 15 through 30: These sections test Basic Page Map logic. The various methods of forming Page Table addresses and RMAs, the Page Map RAMs, and Page Map control are checked.
- o Sections 31 through 41: This area concentrates on the remaining Cache related items. Functions such as Block Fill, Lookahead, and Cache Invalidate are tested.
- o Sections 42 through 44: Final Page Map testing is performed. For example, Page Table Update and the remaining Page Map parity circuits.
- o Sections 45 through 51: This last portion of LMT3/P is devoted primarily to timing and logic conflicts.

All processor activity (micrand execution) is controlled from a previously tested PPU. Controlling signals are sent to and received from the processor by way of the Maintenance Channel and the Maintenance Access Channel (MAC).

A typical test sequence would be:

1. Load the Control Store memory with diagnostic micrands.
2. Load processor registers with operand patterns.
3. Send processor a Start command.
4. The processor now executes the diagnostic micrand or micrand sequence and halts.



#### Typical Test Sequence (continued):

5. The PPU senses halt and directs MAC, via PPU resident code, to read test results from the processor registers.
6. The data retrieved is examined for failure.

All sections, subsections, and conditions should be executed in sequential order. Successful completion is predicated on the successful completion of previously executed sections, subsections and conditions.

#### Repeat Iteration

Implementation of repeat iteration code can, in some subsections, cause displays that have data for conditions that have not been executed. This occurs because these conditions are skipped in order to provide a status report that is normally reported in a later condition. An example of this is in section 30, subsection 1. An error report from iteration 0 consists of reports for conditions 0, 1, 2, and 12. The data displayed for conditions 3 through 11 is residual and is not relevant. Subsection descriptions should be carefully read to determine which condition reports are valid for the iteration error being displayed.

Unless specifically stated in the subsection descriptions, all repeat iterations (SRI) are to condition C.0.

#### Section/Subsection Order Dependencies

LMT3/P is organized into section, subsection and condition elements. The successful completion of any element is a prerequisite for the successful completion of all predecessor elements, therefore the sequential order of execution is encouraged.

Failures detected when test elements are executed out of order should not be used for hardware isolation analysis.

#### CPU Status Reporting:

The CPU Status reported by LMT3/P consists of:

- All Local Memory (LM) parity error status logged in the Processor Fault Status (PFS) registers.
- Parity error status logged in the PFS registers by units adjacent to LM. This data provides additional error isolation.
- Special conditions, either sensed by LMT3/P or extracted from the Monitor Condition register (MCR).

60000285 A

LMT3/P - 2

This information is formatted into two 64-bit words labeled CPU Status Words. One and Two.

- CPU Status Word 1 corresponds, bit for bit, with PFS Register 82. Bits not used by LMT3/P are discarded.
- CPU Status Word 2 represents various status registers:
  - Bits 0 through 11 correspond to PFS Register 83, bits 0 through 11.
  - Bits 12 through 15 are from the Central Memory Uncorrected Error Log 1 (UEL 1) register bits 44 through 47. They represent the RMA Parity Error Code.
  - Bit 16 is generated within LMT3/P and represents CPU time out.
  - Bit 17 is set if the CPU halts at a Control Store address other than the normal halt.
  - Bit 21 is from the Monitor Condition Register and represents Page Search Without Find.
  - Bit 22 is from the Monitor Condition register and represents External Interrupt.
  - Bits 24 through 31 correspond to PFS Register 84, bits 48 through 55.
  - Bits 34 through 39 correspond to PFS Register 80, bits 2 through 7.
  - Bits 40 through 47 correspond to PFS Register 80, bits 18 through 23.
  - Bits 48 through 55 correspond to Central Memory Uncorrected Error Log 1 (UEL1), Register A4, bits 48 through 55.
  - Bits 56 through 63 contain the number of the failing condition, if applicable.

CPU Status is fetched from its source registers at the completion of every CPU operation. A few exceptions exist where timing must be considered. After the status information is fetched it is either:

- Formatted and reported regardless of error. This method is used only when the contents of the status words are explicitly important to the current test sequence, for example when testing parity error sensing logic.
- Checked for error; if none exists the status is discarded. If an error is sensed and it is the first CPU Status error to occur in the current subsection, the Condition Error Flag is set, the status is formatted as shown, the current condition number added to Word Two, and both words saved. If subsequent CPU Status errors occur during this same subsection in a later condition the Condition Error Flag is set but the status information is discarded. Therefore, if the condition data shows the expected and received operands to be the same, yet the Condition Error Flag is set, a CPU Status Error must have occurred. The CPU Status from the first status error, which had been saved, is reported by the last two conditions of the subsection. If no status errors occurred these conditions are left zero.

60000285 A

LMT3/P - 3

## NOTE

Throughout this document condition data is reported as information only. Yet, if a CPU Status Error should occur, the condition Error Flag would be set. The data reported by the condition is still for information only.

### Logic Terminology:

The logic terminology in this document has been updated to match the terminology used in Publications diagrams (levels 1 through 4). The terms given in parentheses following the logic names (for example, IITG0) are those used in the LMT3/P listings (PP and microcode) which have not been updated. Therefore, when in doubt about which terminology is correct (this document or the test code), this document terminology should be assumed to be correct.

### Documentation Conventions:

The following paragraphs define documentation conventions used in LMT3/P section and subsection descriptions.

#### 1. Processor Modes.

Special processor operational modes are selected within LMT3/P. When a mode is selected or deselected it is stated as Select XYZ mode or Deselect XYZ mode. The actual method of selecting that mode is not given. The various modes used and actual methods are listed below.

- FAKECM Mode - Selected by setting bit 61 in the processor Dependent Environment Control (DEC) register. It is deselected by clearing the bit.
- Lookahead Mode - Lookahead (unconditional) is selected by setting bit 50 in the processor DEC register. Conditional Lookahead is selected by also setting bit 51. Lookahead is deselected by clearing bit 50. Conditional Lookahead is deselected by clearing bit 51 and leaving bit 50 set.

#### 2. Format of Variables.

Variables of 64 bits shown within the LMT3/P documentation are shown either fully expanded, such as, 0000 1111 2222 3333(HEX), or in an abbreviated form, such as 0--0 or DF--DF(HEX). The abbreviated variables are also 64 bits but have repeating character strings. If fewer than 64 bits are shown the variable is always right justified.

#### 3. Register File References.

The Register File is used throughout the test to hold operands and results. When referring to specific Register File locations the term RFxx is normally used. The xx refers to the absolute location in hexadecimal (0 through 3F(HEX)). For example RF12 and RFF refer to Register File locations 12(HEX) and F(HEX) respectively.

#### 4. Variables Dependent on CM Configuration.

It is noted in several subsections that some variables (for example, register contents, expected results, etc.) depend on the CM configured with the system being tested. The value of the variables shown throughout this document reflect a 128 Megabyte system. The noted variables are masked to the CM size of the system being tested. The mask is from the Central Memory Options Installed Register.

For example:

| CM Size | Mask (hexadecimal) |
|---------|--------------------|
| 128 MB  | FFFFFF             |
| 96      | BFFFFF             |
| 64      | 7FFFFFF            |
| 32      | 3FFFFFF            |
| 16      | 1FFFFFF            |
| 14      | 1BFFFF             |
| 12      | 17FFFF             |
| 10      | 13FFFF             |
| 8       | FFFFFF             |
| 7       | DFFFF              |
| 6       | 8FFFF              |
| 5       | 9FFFF              |
| 4       | 7FFFF              |
| 3       | 5FFFF              |
| 2       | 3FFFF              |
| 1       | 1FFFF              |

The mask is positioned as required for the particular variable and the correct operand or result is formed.

#### 5. Live Register Reads and Writes.

The documentation makes no distinction between processor Live registers (those requiring micrand assist to access) and other processor registers (those accessed directly by MAC). This note advises the user that a statement such as; set PTA to 0, requires a micrand and the processor must be started to accomplish the task. The Live registers used by LMT3/P are:

PTA - Page Table Address - Written to.  
PTL - Page Table Length - Written to.  
PSM - Page Size Mask - Written to.  
MCR - Monitor Condition Register - Read from.  
UCR - User Condition Register - Read from.

#### 6. Micrand Definitions.

A list of micrands used within a subsection is included in the documentation for each subsection. A short descriptive phrase is used to briefly highlight the micrands purpose. A more complete description of each micrand is given in Appendix B of this document.

#### Special Procedures:

##### 1. Start Processor.

Start processor at Micrand X, states that the CPU is started at a particular CS micrand. Prior to starting the processor the functions Master Clear and Clear Errors are sent unless explicitly stated otherwise.

##### 2. Wait CPU Halted.

The Wait for CPU halted comment specifies one of two situations:

- The processor completes the current micrand sequence and sets the CPU Halted FF.
- The processor is hung and the Wait subroutine has timed out. This will normally be reported as a CPU Status Error and is logged in CPU Status Word Two (bit 16).

##### 3. SVA Addressing.

SVA addressing is always used unless otherwise stated. This is selected by having bit 36 in the Segment Map area of the Functional Unit (FU) micrand set. Instruction Fetch (IF) does not use a micrand to send a request. It, therefore, must use PVAs. See next item for IF procedures.

#### 4. Setting the P Descriptor Register.

When IF is used to generate a request to LM, a PVA consisting of a 32-bit Byte Number (BN) and a 12-bit Segment Number (always zero in LMT3/P) is used. In the Segment Map the 12-bit Segment Number is dropped and a 16-bit ASID, from the P Descriptor register is attached before being sent to LM. The value of the ASID can be changed by changing the contents of the P Descriptor register. The following procedure is used to load the P Descriptor register:

- Both Segment Map sets are disabled.
- The Segment Table Address (STA) register is set to zero.
- RF11 is set to zero.
- Central Memory (CM) address 0 is set to the value to be loaded into the P Descriptor register, formatted as:  
AA00 xxxx 0000 0000(HEX).  
xxxx is the desired ASID value.
- The processor is started at Micrand 10. The Segment Map function code in SMMIC is set to 7 (exchange) and RDSa is set to RF11 (PVA). Because the Segment Map sets are off, a Map Miss occurs and a new Segment Descriptor is loaded from the Segment Table at STA + RF11 (0+0).
- The new Segment Descriptor from CM(0) is sent to the Segment Map where it is entered into the P Descriptor register.
- All PVAs now sent from IF will use the new ASID until it is changed by repeating the same sequence.

#### 5. Instruction Fetch (IF) Usage.

All IF Requests are initiated by setting the desired address (PVA) into RF10 and starting processor at micrand 13. There are some micrand sequences which also initiate IF Requests and they also use RF10. RF10 is copied from the Register File to the live P Register. The processor immediately issues an IF Request using the new P value. The following sequence occurs:

- The 64-bit word at the specified PVA (after PVA to SVA to RMA conversion) is fetched and sent to the IF Unit.
- The upper byte (byte 0) is extracted and 100(HEX) is added to it to form the next Control Store (CS) address. For example, if F0--F0(HEX) is fetched from CM or Cache, 1F0(HEX) is formed and sent to CS.
- The micrand at the specified CS address is executed.

During LMT3/P initialization, CS addresses 100(HEX) to 1FF(HEX) are set to the same micrand. The micrand copies its 64-bit FU portion to RF14. The format of the FU micrand is A5--A5 xyxy(HEX), where xy is the CS address of the micrand minus 100(HEX). Therefore, by checking RF14 it is possible to determine the byte 0 value of the word fetched by IF.

An exception to this method exists during some purge tests. The virtual instruction mechanism must be used to test some purge functions so a purge micrand (micrand 28) is written to CS at one specific address. When the test is complete the CS address is reset to its original contents.

#### 6. Filler Constants.

Prior to starting the processor, all locations to be written by the executed micrand are prefilled with the constant DF--DF(HEX). This provides positive proof that the location was or was not entered.

#### 7. Trap and Interrupt Handling.

All microcode traps and interrupts, should they occur during processor microcode execution, are forced to Control Store (CS) addresses 40 through 47. The micrands at these addresses are designed to set the CPU halted and loop at that address indefinitely.

#### Test Initialization:

- Clear the Processor Test Mode register.
- Clear the Central Memory, IOU, and Processor errors.
- Load Control Store with micrands.
- Disable IOU Test Mode register.
- Set an I4 flag to 1 for later use if needed.
- Set the IOU Test Mode Register to 00-01F(HEX).
- Set the OS Bounds Register to max.
- Clear bit 60 of the IOU DEC Register.
- Put the CPU into C180 and Monitor Mode.
- Enable Central Memory SECEDED.
- Determine CM and Cache configuration.
- Set STA to zero.
- Disable both Segment Maps.
- Clear C180/C170 Decode RAMs.
- Deselect FAKECM mode.
- Deselect Lookahead mode.
- Load Address Control Soft Control Memories with data from Code Set A (similar to product set code).
- Deselect all Cache and Map Sets.
- Clear the P3 DEC register except for the following...
  - Set bit 47 (disable PDM).
  - Retain bit 34 (first failure enable) if set.
  - Retain bits 48 and 49 (margins) if set.
- Set SIT and PIT to FFFFFFFF(HEX).
- Clear the Monitor Mask, User Mask, User Condition register, Monitor Condition register, Segment Table Address, PTA/PTL registers, and the Bounds register.
- Attempt to clear the AC address lines by issuing a Purge AC errors micrand (micrand 5).
- Attempt to clear the free running counter with micrand 4.
- Clear any status errors.
- Initialize 1MB CM to C3C3C30xxxxx where xxxxx = 0-1FFFF(HEX).
- Purge all maps and all cache.

60000285 A

LMT3/P - 8

#### SECTION DESCRIPTIONS

#### SECTION 0 - RMA ADDRESS PATH

Test those areas of Local Memory (LM) where the Real Memory Address (RMA) can be used.

Subsection 0 - Page Size Mask.

Subsection 1 - STR Mode Tag Bits and CMC Tag to CMC Path.

Subsection 2 - CMC Free Running Counter.

Subsection 3 - CMC Refresh and Interrupt Functions.

Subsection 4 - RMA Bank Select/CM Busy.

#### Method:

All subsections within this section use micrands with RMA addressing selected. The individual method used is unique to each subsection and is explained in the subsection documentation.

#### Initialization:

- Set CM address 0-FFFF to C3--C300xxxx(HEX); where xxxx equals the CM word address.

#### Section 0, Subsection 0

Test the Page Size Mask.

This is a preliminary Page Size Mask (PSM) test to verify logic within the PSM network. Final PSM testing occurs during the Hash Network test in a subsequent section.

#### Hardware Tested:

- PSM logic network.

#### Method:

Vary the PSM register from 7F(HEX) to 0 and use Read Word from RMA micrands to fetch words from CM addresses 0, 3FFF(HEX), 2AAA(HEX), and 1555(HEX). The RMA sent to CM should not be affected by the value of the PSM register. If an error occurs using a particular PSM value, the RMA generated will fetch data from the wrong CM address.

Eight passes are required through all conditions to exercise all PSM values.

60000285 A

LMT3/P - 9

Test Patterns (hexadecimal):

| Pass | PSM Value |
|------|-----------|
| 1    | 7F        |
| 2    | 7E        |
| 3    | 7C        |
| 4    | 78        |
| 5    | 70        |
| 6    | 60        |
| 7    | 40        |
| 8    | 00        |

Micrands Used:

| Number | Description         |
|--------|---------------------|
| 1      | Read Word from RMA. |

Initialization:

- Set current PSM value to 7F(HEX).
- Set pass count to 1.

Conditions:

C.0 Pass Count (1 through 8).

- a. Report count; informational.

C.1 Current PSM value.

- a. Enter current value into PSM register.
- b. Report current PSM value; informational.

C.2 Address 0 Test.

- a. Set RF2 to 0--0 (RMA).
- b. Set RF6 to DF---DF(HEX); filler.
- c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF6, report data; error if not C3---C3000000(HEX).

C.3 Address 1FFF(HEX) Test.

- a. Set RF2 to 0--01FFF8(HEX) (RMA).
- b. Set RF6 to DF---DF(HEX); filler.
- c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF6, report data; error if not C3---C3001FFF(HEX).

Conditions:

C.4 Address 2AAA(HEX) Test.

- a. Set RF2 to 0-15550(HEX) (RMA).
- b. Set RF6 to DF---DF(HEX); filler.
- c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF6, report data; error if not C3---C3002AAA(HEX).

C.5 Address 1555(HEX) Test.

- a. Set RF2 to 0--0AAA8(HEX) (RMA).
- b. Set RF6 to DF---DF(HEX); filler.
- c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF6, report data; error if not C3---C3001555(HEX).

C.6 CPU Status Word 1.

- a. Report word 1 if saved; error if reported.

C.7 CPU Status Word 2.

- a. Report word 2 if saved; error if reported.

C.8 First Failure Capture Information.

- a. Report PFS87 bits 19-23 in bits 59-63; information only.
- b. Report last PTL written in bits 34-47; information only.
- c. Report last PTA written in bits 13-31; information only.

Exit subsection if eight passes complete; else shift PSM and return to C.0.

Section 0, Subsection 1

This subsection tests STREAM (STR) MODE TAG BITS (IITGO) and CMC TAG BITS to CMC (LMTAG) path.

Test the LMTAG by way of the IITGO path. The IITGO path comes from OPI and is used to define the destination register when Stream Mode is invoked.

Hardware Tested:

- LMTAG Generation (CMC TAG GENERATOR).
- Exchange Jump Control.



# Method:

The IITGO path (OPI to LM) and the LMTAG path (LM to CMC) are tested by selecting Stream Mode in the processor. When in Stream Mode the 8-bit Register File address is sent with the read or write function and is then passed on to CMC as part of the request tag. In this way the lower six bits of the Tag can be verified by sensing which Register File address received the data from CM.

Stream Mode is selected and single reads are performed. The CM address and the Register File address used are equal. That is, data from CM(3F(HEX)) is read to Register File address 3F(HEX). CM data has been initialized to reflect its CM address. Test addresses used are 0, 3F(HEX), 15(HEX), and 2A(HEX).

Conditions 8 and 9 are unique in that data is actually streamed from CM to the Register File. Twenty words are read from CM 10(HEX) to 23(HEX) to like addresses in the Register File.

## Test Patterns:

Refer to Conditions.

## Micrands Used:

| Number | Description                                           |
|--------|-------------------------------------------------------|
| 1      | Read Word from RMA.                                   |
| 2      | Read Word from RMA (Stream Mode)                      |
| 32     | Stream Multiple Words from RMA to; RF10 through RF1F. |
| 33     | Stream Multiple Words from RMA to; RF20 through RF2F. |

## Initialization:

- Set PSM to 7F(HEX).

## Conditions:

- C.0 Register file destination address 00).
- Report 0s; informational.
- C.1 Test 0s.
- Set RF2 to 0--0 (RMA).
  - Set RF0 to DF--DF(HEX); filler.
  - Start processor at micrand 2, stream one word from CM, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF0, report data; error if not C3--C30000(HEX).
- C.2 Register file destination address 3F(HEX).
- Report 3F(HEX); informational.

60000285 A

LMT3/P - 12

## Conditions:

- C.3 Test 1s.
- Set RF2 to 0--01F8(HEX) (RMA).
  - Set RF3F to DF--DF(HEX); filler.
  - Start processor at micrand 2, stream 1 word from CM, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF3F, report data; error if not C3--C3003F(HEX).
- C.4 Register file destination address 15(HEX).
- Report 15(HEX); informational.
- C.5 Test odd bits.
- Set RF2 to 0--0A8(HEX) (RMA).
  - Set RF15 to DF--DF(HEX); filler.
  - Start processor at micrand 2, stream 1 word from CM, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF15, report data; error if not C3--C30015(HEX).
- C.6 Register file destination address 2A(HEX).
- Report 2A(HEX); informational.
- C.7 Test even bits.
- Set RF2 to 0--0150(HEX) (RMA).
  - Set RF2A to DF--DF(HEX); filler.
  - Start processor at micrand 2, stream 1 word from CM, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF2A, report data; error if not C3--C3002A(HEX).
- C.8 Register file final destination address 23(HEX).
- Report 23(HEX); informational.
- C.9 Test multiple location stream.
- Set RF2 to 0--080(HEX); RMA, word address 10(HEX).
  - Set RF3 to 0--0F(HEX); word count (HEX).
  - Set RF10 through RF23 to DF--DF(HEX); filler.
  - Start processor at micrand 32, stream words from CM, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 100(HEX); RMA, word address 20(HEX).
  - Set RF3 to 0--04(HEX); word count.
  - Start processor at micrand 33, stream words from CM, wait for CPU halted.
  - Read and check each destination register for expected data. If an error is detected, report the register number, and exit the condition. Error if register contents not C3--C30000xx; where xx is 10(HEX) through 23(HEX). Report data from last Register File location checked; error if not C3--C3000023(HEX).
  - Read CPU status, format and save if error; error if not 0.

60000285 A

LMT3/P - 13

Conditions:

- C.10 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Section 0, Subsection 2

This subsection tests the CMC Free Running Counter.

Test the Free Running Counter (FRC) for basic functional use. Ensure it can be modified from the IOU, that it can be read from the CPU, and that it counts at a one microsecond rate.

Hardware Tested:

- Function Code Receivers.
- Function Code Transmitters.
- Free Running Counter.

Method:

The Free Running Counter (FRC) can only be written to (modified or cleared) from the IOU and can only be read from the CPU. Therefore, the test sequence is to write a pattern to the counter from the IOU, then start the CPU and read the contents to the Register File. Prior to writing the FRC from the PRU, the CPU is started, the Central Memory is resynced and the FRC read. This resync/read acts to synchronize the CPU and the PPU.

Various patterns are used to check the counters ability to hold data: 0s, Fs, 5s, and As. The low order 4 bits cannot be correctly predicted for a given time period. Therefore, the FRC counter will be read 17 times and a check will then be made for the correct pattern within one of those reads. The results for the lower 4 bits will then be put into the actual data buffer as if it was read on the first read. The FRC counter has a possibility of skipping a count. Therefore, a count that is one more than expected will be acceptable.

A check will also be made to ensure that the free running counter is counting at a one microsecond rate and that the counter can only be read after it has changed to a new value.

Conditions 14 and 15 test to ensure that central memory refresh works properly when the FRC register is repetitively read.

NOTE

This subsection will fail if the frequency switch on the IOU is not in the normal position. Also, if the memory margins are set to other than normal, conditions other than zero will be omitted.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description                                                              |
|--------|--------------------------------------------------------------------------|
| 4      | Read the Free Running Counter to 17 Consecutive register file locations. |

Initialization:

- None.

Conditions:

- C.0 Data to be sent to counter (1s).
  - a. Report 0000FF--FF(HEX); informational.

Exit subsection if the central memory margins are not on normal, as determined by the Central Memory Status Summary Register.

- C.1 Test for 1s pattern.
  - a. Set RF2 to 0--0; RMA, not used.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 4, resync Central Memory, read FRC, wait for CPU halted.
  - d. Write pattern 0000FFFFFFFFF4B(HEX) to FRC.
  - e. Start processor at micrand 4, read FRC, wait for CPU halted.
  - f. Read CPU status, format and save if error; error if not 0.
  - g. Read RF6, report data; error if not 0000FF--Fx(HEX), or report error if x does not equal F or E as read from RF6 through RF16.

Conditions:

- C.2 Test for 0s pattern.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 4, resync Central Memory, read FRC, wait for CPU halted.  
c. Write pattern 0000FFFFFFFFF4E(HEX) to FRC.  
d. Start processor at micrand 4, read FRC, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF6, report data; error if not 0000--0x(HEX), or report error if x does not equal 0 or 1 as read from RF6 through RF16.
- C.3 Data to be sent to Counter (5s).  
a. Report 000055--55(HEX); informational.
- C.4 Read and check 5s.  
a. Start processor at micrand 4, resync Central Memory, read FRC, wait for CPU halted.  
b. Write pattern 00005555555554A3(HEX) to FRC.  
c. Start processor at micrand 4, read FRC, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 000055--5x(HEX), or report error if x does not equal 5 or 6 as read from RF6 through RF16.
- C.5 Data to be sent to Counter (As).  
a. Report 0000AA--AA(HEX); informational.
- C.6 Read and check As.  
a. Start processor at micrand 4, resync Central Memory, read FRC, wait for CPU halted.  
b. Write pattern 0000AAAAAAAAA9F8(HEX) to FRC.  
c. Start processor at micrand 4, read FRC, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0000AA--Ax(HEX), or report error if x does not equal A or B as read from RF6 through RF16.
- C.7 Data to be sent to counter (01's).  
a. Report 0000 0101 0101 0101(HEX); informational.
- C.8 Read and check 01's.  
a. Start processor at micrand 4, resync Central Memory, read FRC, wait for CPU halted.  
b. Write pattern 0000 0101 0101 004F(HEX) to FRC.  
c. Start processor at micrand 4, read FRC, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0000 0101 0101 010X(HEX), or report error if X does not equal 1 or 2 as read from RF6 through RF16.

Conditions:

- C.9 Check counters increment rate.  
a. Start processor at micrand 4, resync Central Memory, read FRC, wait for CPU halted.  
b. Write 0--0 to FRC.  
c. Large delay.  
d. Start processor at micrand 4, read FRC, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF6, report data; error if not 0-0501xx(HEX), where xx = 00-FF(HEX).
- C.10 Test the counters ability to lock out reads until value has changed.  
a. Start processor at micrand 4, read FRC, wait for CPU halted.  
b. Format the lower 4 bits of RF6 through RF15 into an actual buffer that can be reported. Discontinue formatting if an error occurred. RF6 is put into bits 0 through 3, RF7 is put into bits 4 through 7, ... until RF15 is put into bits 60 through 63.  
c. Report an error if RF7 is not 1 or 2 more than RF6, RF8 is not 1 or 2 more than RF7, ... until done.  
Sample expected: 679A BCDE F013 4567(HEX).
- C.11 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.13 RF address read and First Failure Capture Information.  
a. Report the RF address, in bytes 6 and 7, that was used in earlier condition reporting.  
b. Report PFS87 bits 19-23 in bits 11-15; information only.
- Conditions 14 and 15 are performed only on iteration 0.
- C.14. Check refresh during FRC register read.  
a. Write CM locations 0 through 100(HEX) with data = 6EC3 A578 0FFB D7E0(HEX).  
b. Disable SECDED by setting bit 1 of the M3 ECR register.  
c. Perform large timeout.  
d. Clear RF02 and set RF06 = DFDF--DF(HEX).  
e. Start processor at micrand 4; wait for CPU halted.  
f. Perform steps c through e 10 times.  
g. Read and compare CM locations 0 through 100(HEX) with data written.  
h. Report CM location that failed the compare if an error occurred; should equal zero.  
i. Enable SECDED by clearing bit 1 of the M3 ECR register.
- C.15 Report CM data read.  
a. Report the expected and actual data for C.14. No error is ever reported in this condition.

### Section 0, Subsection 3

This subsection tests the Interrupt and Refresh Counter resync.

Test the Interrupt function when issued from either the IOU or the CPU. Also test the ability of the CPU to resync the Central Memory Refresh Counter.

#### Hardware Tested:

- Function Code Receivers.
- Function Code Transmitters.
- Refresh network in CMC.

#### Method:

Condition 0 tests the Refresh Counter Resync Function to CMC. This function is tested by recording the time required between two Resync functions. A Resync function is issued followed by a read of the CMC Free Running Counter (FRC). The FRC is sampled only at Bank 0 time. A second Resync is issued and again followed by a read of the FRC. The second FRC read must wait for refresh to resync before Bank 0 can be used. If the function is not correctly received by CMC, the time between the first FRC read and the second FRC read will be significantly less than expected.

Conditions 3 and 6 test the External Interrupt function. The function is tested from both the IOU and from the processor. An Interrupt function is sent from the IOU. The Monitor Condition register (MCR) is read to determine if the Interrupt bit is set. When testing from the CPU a micrand is issued with an Interrupt function code. The Processor to Interrupt Code (LMA) is defined in the RDSc micrand field. The MCR is read to determine that the interrupt was sensed.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                           |
|--------|---------------------------------------|
| 6      | Refresh Counter Resync Test Sequence. |
| 7      | Interrupt function.                   |

#### Initialization:

- None.

#### Conditions:

- C.0 If running on CPU-1, go to C.3.  
Report CPU time Between Resync Functions.  
a. Clear the RFC.  
b. Start processor at micrand 6, test Resync function, wait for CPU halted.  
c. Report execution time in microseconds, expect E, F, or 10(HEX).
- C.1 Report CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.2 Report CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.3 Test Interrupt from IOU.  
a. Start processor at Micrand 0, do not wait for CPU halted.  
b. Send Interrupt Function to processor A.  
c. Wait for CPU halted.  
d. Read CPU Status; error if Interrupt bit in MCR not set. Report Interrupt bit (0--01).
- C.4 Report CPU Status Word 1.  
a. Report word 1; error if not 0.
- C.5 Report CPU Status Word 2.  
a. Report Word 2; error if not 0000820--0(HEX); CPU Time Out and Interrupt.
- C.6 Test Interrupt from processors.  
a. Start processor at micrand 7 if running on CPU-A, or at micrand 65 if running on CPU-B, Interrupt function, wait for CPU halted.  
b. Read CPU status, format and save if error; error if MCR Interrupt bit not set and all else not clear.  
c. Report 1 if Interrupt set; else 0.
- C.7 CPU Status Word 1.  
a. Report word 1; error if reported.
- C.8 CPU Status Word 2.  
a. Report word 2; error if not 0000820--0(HEX); CPU Time Out and Interrupt.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

## Section 0, Subsection 4

This subsection checks CM bank bit selection, and CM busy.

### Hardware Tested:

- CM Bank Select Network.
- CM Busy Logic.

### Method:

CM bank bit selection is tested by initializing all of CM up to the minimum configuration) with a counting pattern. Noninterleaved mode is then selected and words are read by the CPU. RMA bits 12, 13, 14 are used to address words initially written using bits 26, 27, 28.

CM busy logic is tested by issuing three write operations in rapid sequence to different addresses within the same bank. If busy does not come up, one of the write operations will be skipped.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description                 |
|--------|-----------------------------|
| 1      | Read Word from RMA.         |
| 8      | Three Writes to three RMAs. |

### Initialization:

- None

### Conditions:

- C.0 Read Data from CM(0).
- Set RF2 to 0--0 (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C3000000(HEX).

### Conditions:

- C.1 Read data from CM(1).
- Set RF2 to 0--008(HEX) (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C300 0001(HEX).
- C.2 Read data from CM(2).
- Set RF2 to 0--010(HEX) (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C300 0002(HEX).
- C.3 Read data from CM(3).
- Set RF2 to 0--018(HEX) (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C300 0003(HEX).
- C.4 Read data from CM(4).
- Set RF2 to 0--020(HEX) (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C300 0004(HEX).
- C.5 Read data from CM(5).
- Set RF2 to 0--028(HEX) (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C300 0005(HEX).
- C.6 Read data from CM(6).
- Set RF2 to 0--030(HEX) (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C300 0006(HEX).

Conditions:

- C.7 Read data from CM(7).  
a. Set RF2 to 0--038(HEX) (RMA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C300 0007(HEX).
- C.8 CM busy test, report CM(0) data.  
a. Set RFA to F0--F0(HEX); write data, first write.  
b. Set RFB to F1--F1(HEX); write data, second write.  
c. Set RFC to F2--F2(HEX); write data, third write.  
d. Start processor at micrand 8, 3 sequential writes, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read CM(0), report data; error if not F0--F0(HEX).
- C.9 Report CM(1) data.  
a. Read CM(1), report data; error if not F1--F1(HEX).
- C.10 Report CM(2) data.  
a. Read CM(2), report data; error if not F2--F2(HEX).
- C.11 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

SECTION 1 - PARITY CIRCUITS USING RMA ADDRESSES

Test all parity circuits where RMA addressing can be used.

Subsection 0 - STR Mode Tag Bits and CMC Tag to CMC Parity.

Subsection 1 - CM Fault Responses (Part 1).

Subsection 2 - CM Fault Responses (Part 2).

Subsection 3 - Page Offset/PSM Parity.

Subsection 4 - RMA Parity Generation Logic.

Subsection 5 - RMA Parity Error Signals.

Method:

Subsections within this section use only RMA addressing. Parity testing is normally divided into two areas. First the verification of the parity checkers and/or generators to ensure no false parity indications are detected or generated. Second, the verification of the parity status signal to MAC. This is normally done by forcing a parity error at the checker or generator followed by a status check to ensure the expected status has occurred.

See individual subsections for additional detail.

Initialization:

- Set CM; 0 through A(HEX) to C3--C300000x. x = 0 through A(HEX).

Section 1, Subsection 0

This subsection tests the CMC tag to CMC (LMTAG) parity generator, and STR mode tag bits (CMTAG) parity checker.

Test the tag parity circuits to ensure false errors are not generated and that parity errors can be detected by MAC.

Hardware Tested:

- LMTAG Parity Generator. ZIF Board BF at B1G.
- CMTAG Parity Checker. ZIF Board BH at B1F.
- PFS82 bit 62. MAC.

#### Method:

Stream Mode is used in this subsection to enable maximum flexibility in the construction of the LMTAG. All possible bit combinations are used within the word number portion of the Stream or Exchange Jump Tag.

To test for false parity error indications, CM is initialized using a counting pattern. Data is then transferred, in Stream Mode, from CM to like locations in the Register File. Status is checked after each block transfer. The same block is then written back to CM and the status is again checked. All blocks are 10(HEX) words long. This procedure is repeated four times using all Register File locations.

The Register File locations holding the test variables, RMA and word count, differ between pass one and the other passes. This is because the locations used by pass one are destroyed by pass two.

The parity error signal to MAC is tested by selecting the CPTAG bit in the processor PTM register. A RMA read is executed and the parity status should be set.

The Exchange Tag PE (PFS82 bit 52) is also tested. When setting PTM bit 44 and executing a RMA read in Stream mode, the PFS82 bit 52 should be set.

#### Test Patterns:

##### CM Data Block (hexadecimal):

| Pass | From       | To         |
|------|------------|------------|
| 1    | C3--C30000 | C3--C3000F |
| 2    | C3--C30010 | C3--C3001F |
| 3    | C3--C30020 | C3--C3002F |
| 4    | C3--C30030 | C3--C3003F |

#### Micrands Used:

| Number        | Description                 |
|---------------|-----------------------------|
| 31 through 34 | Stream Words from CM to RF. |
| 35 through 38 | Stream Words to CM from RF. |
| 1             | Read word from RMA.         |

#### Initialization:

- Set PSM to 7F(HEX).
- Set all configured CM to a special pattern.
- Set Pass Count to 1.

#### Conditions:

- C.0 Pass Count (1 through 4).
- Report pass count; informational.
- C.1 Read 10(HEX) words, report last CM address read.
- If pass 1, set RF12 to 0--0; starting RMA.
  - If pass 2, set RF2 to 0--080(HEX); starting RMA.
  - If pass 3, set RF2 to 0--0100(HEX); starting RMA.
  - If pass 4, set RF2 to 0--0180(HEX); starting RMA.
  - Set DF--DF(HEX) into registers to be written.
  - If pass 1, set RF13 to 0--0F(HEX); word count.
  - If pass 2-4, set RF3 to 0--0F(HEX); word count.
  - Start processor at micrand 30 plus pass count, read 10(HEX) words from RMA, Stream Mode, wait for CPU halted.
  - Read Register File locations just filled; error if not the same as CM data. If error, report data and exit condition; else report last location tested.
- C.2 CPU Status Word 1.
- Read CPU status, format and report word 1; error if not 0.
- C.3 CPU Status Word 2.
- Report word 2; error if not 0.
- C.4 Write 10(HEX) words, report last CM address written.
- Set CM block to be written to DF--DF(HEX).
  - Start processor at micrand 34 plus pass count, write 10(HEX) words to RMA, Stream Mode, wait for CPU halted.
  - Read CM addresses just written; error if not the same as Register File data. If error, report data and exit condition; else report last location tested.
- C.5 CPU Status Word 1.
- Read CPU status, format and report word 1; error if not 0.
- C.6 CPU Status Word 2.
- Report word 2; error if not 0.
- Continue to next condition if error, Scope Mode Set (SSM), or if 4 passes complete; else return to C.0.
- C.7 Test Tag Parity Error signal, report CPU Status Word 1.
- Set RF2 to 0--0 (RMA).
  - Set CPTAG bit; bit 28 in the processor PTM register.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Read CPU status, format and report word 1; error if not 0--02 (HEX), CMTAGPE set.
- C.8 CPU Status Word 2.
- Report word 2; error if not 00030---0(HEX).
  - Clear bit 28 in processor PTM register.

Clear bit 28 in the processor PTM register.



Conditions:

- C.9 Test Exchange Tag PE, report CPU status word 1.
- Set RF2 to 0---0 (RMA).
  - Set PTM bit 44.
  - Start processor at micrand 2, read word from RMA in Stream mode, wait for CPU halted.
  - Read CPU status, format and report status word 1; error if not 0---0800(HEX).
- C.10 CPU status word 2.
- Report status word 2.
- C.11 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Clear PTM bit 44.

Section 1, Subsection 1

This subsection tests the CM fault response codes (part 1).

Test four of the six error responses sent from CMC.

Hardware Tested:

- CM Response Translator.
- PFS82 bits 56 through 59.
- CM Response Transmitters.

Method:

Each response bit tested is treated individually. The procedure is described below. Three conditions are required for each bit; one to report the data read or written and two to report the CPU status.

- C.0 through C.2 Reject.
- C.3 through C.5 Response Code Parity Error.
- Select the Complement Response Code Parity bit (bit 29) in the processor PTM register and issue a read operation.
- C.6 through C.8 Uncorrected Write Error.
- Set the Central Memory Bounds register and issue a write operation to an address outside the bounds.

60000285 A

LMT3/P - 26

Method (continued):

- C.9 through C.11 Uncorrected Read Error.
- Set a word into CM containing an ECC to create a double bit error and issue a read operation.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description         |
|--------|---------------------|
| 1      | Read Word from RMA. |
| A      | Write Word to RMA.  |

Initialization:

- Set CM0 through FF(HEX) to C3--C30000xx(HEX). xx = 0 through FF (HEX).

Conditions:

- C.0 Reject response read data.
- Set RF2 to 0--0 (RMA).
  - Set RF6 to DF--DF(HEX); filler.
  - Set the CPFUNC bit, bit 34, in the processor PTM register.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Clear CPFUNC bit in processor PTM register.
  - Read RF6, report data; error if not C3-C3000000(HEX).
- C.1 CPU Status Word 1.
- Read CPU status, format and report word 1; error if not 0000080--08(HEX), (SELRD(0), CMRJCT).
- C.2 CPU Status Word 2.
- Report word 2; error if not 000100FF30000000(HEX).
- C.3 If running on CPU-B, go to C.6.
- Response code PE read data.
- Set RF6 to DF--DF(HEX); filler.
  - Set the Complement Response Code Parity Bit (bit 29) in the processor PTM register.
  - Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - Clear bit 29 in processor PTM register.
  - Read RF6, report data; error if not C3--C3000000(HEX).

60000285 A

LMT3/P - 27

# Conditions:

- C.4 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0--04 (HEX) (CMRSPE).
- C.5 CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.6 Uncorrected write error, report CM data.
  - a. Set RFA to F7--F7(HEX); write data.
  - b. Read Central Memory Bounds register (register 21), save contents.
  - c. Write 80--0(HEX) or 20--0(HEX) if running on CPU-B into Bounds register (selects upper and lower address of 0).
  - d. Start processor at micrand A, write word to RMA, wait for CPU halted.
  - e. Restore original data to Bounds register.
  - f. Read CM(0), report data; error if not C3--C3000000(HEX); write is blocked.
- C.7 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0--020 (HEX) (CMWTUN).
- C.8 CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.9 Uncorrected read error, report read data.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Set the Central Memory Environment Control register to 1000 0000 0000 0000(HEX); write check bits.
  - c. Set CM(0) to 3A0--0(HEX); double bit error data.
  - d. Clear Central Memory EC register.
  - e. Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - f. Read RF6, report data; error if not 3A0--0(HEX).
- C.10 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0000080--010(HEX), (SELRD(0), CMRDUN).
- C.11 CPU Status Word 2.
  - a. Report word 2; error if not 000000FF30-0(HEX).
  - b. Set CM(0) to C3--C30000(HEX); restore good data.
- C.12 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

## Section 1, Subsection 2

This subsection tests the CM fault response codes (part 2).

Test the remaining two CMC Fault Response Codes. Also, test two areas in CMC not testable from the IOU; Suppress Single Bit Error Detection and Processor Port Disable.

### Hardware Tested:

- CM Response Translator.
- PFS82 bits 80 and 81.
- CM Response Transmitters.

### Method:

The two response bits are tested as follows.

- C.0 through C.2 Corrected write error.  
Data containing an ECC to produce a single bit error is set into CM. A partial write is executed.
- C.3 through C.5 Corrected read error.  
The same data used during the corrected write test and left in CM is now read.

The remaining two tests are formatted in a similar manner.

- C.6 through C.8 Suppress single bit detection.  
Set the suppress bit (bit 38) in the Central Memory EC register. Again read CM(0), which contains the data designed to produce a single bit error. No error should be detected.
- C.9 Processor Port Disable.  
Set the disable bit (bit 32) for processor A, in the Central Memory EC register, read good data from the Central Memory. The CPU should hang waiting for the CMC response.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| 1      | Read Word from RMA. |
| A      | Write Word to RMA.  |
| B      | Write Byte to RMA.  |

#### Initialization:

- Set CM(0) to C3--C3000000(HEX).

#### Conditions:

- C.0 Corrected write error write data.
- a. Set RF2 to 0--0 (RMA).
  - b. Set RF12 to 0--0; byte count, one byte.
  - c. Set RFA to 0--0F0(HEX); write data.
  - d. Set the Central Memory EC register to 1000 0000 0000 0000(HEX); allows write of CM data with bad ECC.
  - e. Set CM(0) to 72850--0(HEX); single bit error data.
  - f. Clear Central Memory EC register.
  - g. Start processor at micrand B, write byte to RMA, wait for CPU halted.
  - h. Read CM(0), report data; error if not F0850--0800(HEX).
- C.1 CPU Status Word 1.
- a. Read CPU status, format and report word 1; error if not 0--080 (HEX) (CMWTCR).
- C.2 CPU Status Word 2.
- a. Report word 2; error if bit 23 (MCR Soft Error) not set and all others not clear.
- C.3 Corrected read error read data.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Set the Central Memory EC register to 100--0(HEX); allows write of CM data with bad ECC.
  - c. Set CM(0) to 72850--0(HEX); single bit error data.
  - d. Clear Central Memory EC register.
  - e. Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - f. Read RF8, report data; error if not 72850--08000(HEX).
- C.4 CPU Status Word 1.
- a. Read CPU status, format and report word 1; error if not 0--040 (HEX) (CMRDCR).
- C.5 CPU Status Word 2.
- a. Report word 2; error if not 0.
- C.6 Suppress single bit detection read data.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Set the Suppress bit (bit 38) in the Central Memory EC register.
  - c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - d. Clear bit 38 in Central Memory EC register.
  - e. Read RF8, report data; error if not 72850--08000(HEX).

#### Conditions:

- C.7 CPU Status Word 1.
- a. Set CM(0) to C3--C3000000(HEX); restore good data.
  - b. Read CPU status, format and report word 1; error if not 0.
- C.8 CPU Status Word 2.
- a. Report word 2; error if not 0.
- C.9 Processor Port Disable Read Data.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Set the processor Port Disable bit (bit 32 CPU-A, bit 34 CPU-B) in the Central Memory EC register.
  - c. Start processor at micrand 1, read word from RMA, wait for CPU halted, or CPU hang time out.
  - d. Clear bit 32 in Central Memory EC register.
  - e. Read CPU status, report CPU Hang Status; error if not set.
- C.10 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

#### Section 1, Subsection 3

This subsection tests the page offset/PSM parity circuits.

Test parity bit generation and parity checking for the Page Offset register. Also, test the detection of a Page Offset (PO) and Page Size Mask (PSM) parity error.

#### Hardware Tested:

- PO Parity Generator.
- PO Parity Checker.
- PSM Parity Checker.

#### Method:

The PO parity generator and checker are tested at the same time by using an incrementing series of RMA addresses. The generator and the checker are each made from two chips, however, the bits are divided differently. The generator is split (bits 48 through 54 and bits 55 through 60), and the checker is split (bits 48 through 56 and bits 57 through 60). To thoroughly test both networks, a counting pattern is used. The number of bits in the largest side of each split governs the count limit. Two counts are required; one for the right side and one for the left. The right side count uses bits 55 through 60; 6 bits or 64 passes. The left side count uses bits 48 through 56; 9 bits or 512 passes. Therefore, 512 plus 64 or 576 passes are required to complete the test.

The parity signal (POPE) to MAC is tested by selecting the Complement Page Offset Parity bit (CPP0) in the processor PTM register. Two operations are executed, one with an even number of bits (zero parity bit) and with an odd number of bits (one parity bit).

The PSM parity signal (PSMPE) to MAC is tested by setting data with wrong parity into the PSM register. This data is used during read operation and the error sensed.

#### Test Patterns (hexadecimal):

| Pass | RMA                                         | Expected Data                     |
|------|---------------------------------------------|-----------------------------------|
|      | increment<br>bits 55 through 80             |                                   |
| 1    | 0----0                                      | C3--C3000000                      |
| 2    | 0----8                                      | C3--C3000001                      |
| 3    | 0---10                                      | C3--C3000002                      |
|      |                                             |                                   |
|      | continue to<br>increment<br>by 8            | continue to<br>increment<br>by 1  |
|      |                                             |                                   |
| v    |                                             |                                   |
| 40   | 0--1F8                                      | C3--C300003F                      |
|      | start<br>increment of<br>bits 48 through 56 |                                   |
| 41   | 0-----0                                     | C3--C3000000                      |
| 42   | 0---080                                     | C3--C3000010                      |
| 43   | 0--0100                                     | C3--C3000020                      |
| 44   | 0--0180                                     | C3--C3000030                      |
|      |                                             |                                   |
|      | continue to<br>increment<br>by 80           | continue to<br>increment<br>by 10 |
|      |                                             |                                   |
| v    |                                             |                                   |
| 240  | 0--FF80                                     | C3--C3001FF0                      |

#### Micrands Used:

| Number | Description         |
|--------|---------------------|
| 1      | Read Word from RMA. |

#### Initialization:

- Clear current RMA.
- Set Pass Count to 1.

60000285 A

LMT3/P - 32

#### Conditions:

- C.0 Pass count (1 through 240(HEX).  
a. Report pass count; informational.
- C.1 Expected read data.  
a. Generate and report the expected data for current pass; informational.
- C.2 Actual read data.  
a. Set RF2 to current RMA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
d. Read RF6, report data; error if not as reported in C.1.
- C.3 CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.4 CPU Status Word 2.  
a. Report word 2; error if not 0.
- Continue to next condition if error, SSM, repeat iteration (SRI), or if 240 HEX) passes complete; else return to C.0.
- C.5 Force POPE using even data; read data.  
a. Set RF2 to 0--0 (RMA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set the CPP0 bit (bit 10) in the processor PTM register.  
d. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
e. Clear bit 10 in processor PTM register.  
f. Read RF6, report data; error if not C3--C3000000(HEX).
- C.6 CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0--02000 (HEX) (POPE).
- C.7 CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.8 Force POPE using odd data; read data.  
a. Set RF2 to 0--08(HEX) (RMA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set the CPP0 bit (bit 10) in the processor PTM register.  
d. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
e. Clear bit 10 in processor PTM register.  
f. Read RF6, report data; error if not C3--C3000001(HEX).
- C.9 CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0--02000 (HEX) (POPE).

60000285 A

LMT3/P - 33

# Conditions:

- C.10 CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.11 Force PSMPE; read data.
  - a. Set RF2 to 0--0 (RMA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Set the MAC Data Output bit (bit 18) in the processor PTM register.
  - d. Set PSM register; Processor Register 4A to 7F(HEX); sets wrong parity in PSM.
  - e. Clear processor PTM bit 18.
  - f. Start processor at micrand 1, read data from RMA, wait for CPU halted.
  - g. Read RF8, report data; error if not C3--C3000000(HEX).
- C.12 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0--01000 (HEX) (PSMPE).
- C.13 CPU Status Word 2.
  - a. Report word 2; error if not 0.
  - b. Clear bad parity data from processor PSM.
- C.14 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

## Section 1, Subsection 4

This subsection tests the real memory address parity checkers.

Test the RMA parity generators to ensure no false errors are sensed and logged.

### Hardware Tested:

- RMA Parity Generator.

### Method:

A repeated counting pattern is used to build the RMA. Because the RMA parity generator is organized into unequal segments (6 bit and 8 bit), some repetition in the 6-bit fields exist.

|        |        |        |        |        |                                          |
|--------|--------|--------|--------|--------|------------------------------------------|
| 2      | 7 8    | 15 16  | 23 24  | 29     | Bit 29<br>is always<br>high (logic zero) |
| 6 bits | 8 bits | 8 bits | 8 bits | 8 bits |                                          |

60000285 A

LMT3/P - 34

Patterns are incremented after each pass (256 required) and the formed RMA is then masked to limit the highest address to the CM configuration of this system.

### Test Patterns (hexadecimal):

| Pass | RMA (shown masked to 128 MB system) | Expected Result  |
|------|-------------------------------------|------------------|
| 1    | 000000                              | C3--C3 C300 0000 |
| 2    | 010108                              | C3--C3 C300 2021 |
| 3    | 020210                              | C3--C3 C300 4042 |
| 4    | 030318                              | C3--C3 C300 6063 |
| ↓    | ↓                                   | ↓                |
| 64   | 3F3FF8                              | C3--C3 C307 E7FF |
| 65   | 404000                              | C3--C3 C308 0800 |
| ↓    | ↓                                   | ↓                |
| 256  | FFFFF8                              | C3--C3 C31F FFFF |
| ↓    | ↓                                   | ↓                |
| 2147 | 7FFFFFF8                            | C3--C3 C3FF FFFF |

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| 1      | Read Word from RMA. |
| A      | Write Word from RM. |

### Initialization:

- Write all RMA from above table with pattern.
- Set current RMA to 0.
- Set Pass Count to 1.

### Conditions:

- C.0 Pass count (1 through 100(HEX).
  - a. Report pass count; informational.
- C.1 Current RMA.
  - a. Set RF2 to current RMA.
  - b. Report current RMA; informational.
- C.2 Expected read data.
  - a. Format and report expected result; informational.

60000285 A

LMT3/P - 35

#### Conditions:

- C.3 Actual read data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
c. Read RF8, report data; error if not as reported in C.2.
- C.4 CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.5 CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.6 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit subsection if all\*\*\* passes complete; else advance pattern and return to C.0.

\*\*\*For: 16 MB - 100(HEX) passes.  
32 MB - 200(HEX) passes.  
64 MB - 400(HEX) passes.  
128 MB - 800(HEX) passes.

#### Section 1, Subsection 5

This subsection tests the real memory address parity signals.

Test the Complement RMA Parity bit signals and the RMA parity error signals to CMC and to the Central Memory Uncorrected Error Log 1.

#### Hardware Tested:

- RMA Parity Status.
- Central Memory Parity Detection.

#### Method:

Wrong parity is forced at the RMA parity generator by setting the complement parity bits (CPRMA) in the processor PTM register. Data is read from selected RMAs causing address parity errors. The status is checked for expected errors.

All 4 bits are tested for functional indication, but only the lower 3 bits are tested individually. The high order bit is beyond the limits of the processor maximum memory configuration.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description         |
|--------|---------------------|
| 1      | Read Word from RMA. |

#### Initialization:

- None.

#### Conditions:

- C.0 Select all status bits, even RMA, report read data.  
a. Set RF2 to 0--0 (RMA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Set the Complement RMA (CPRMA) bits 30 through 33 in the processor PTM register.  
d. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
e. Read RF8, report data; error if not C3--C3000000(HEX).
- C.1 CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0000080--010(HEX) (SELRD(0), CMRDUN).
- C.2 CPU Status Word 2.  
a. Report word 2; error if not 000400FF30--0(HEX).
- C.3 Select all status bits, odd RMA, report read data.  
a. Set RF2 to 0--020220(HEX) (RMA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
d. Read RF8, report data; error if not C3--C3000000(HEX).
- C.4 CPU Status Word 1.  
a. Clear bits 30-32 in processor PTM register.  
b. Read CPU status, format and report word 1; error if not 0000080--010(HEX) (SELRD(0), CMRDUN).
- C.5 CPU Status Word 2.  
a. Report word 2; error if not 000400FF30--0(HEX).

Conditions:

- C.6 Select status bit 0.
  - a. Set RF2 to 0--0 (RMA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - d. Clear bit 33 in processor PTM register.
  - e. Read RF8, report data; error if not C3--C3000000(HEX).
- C.7 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0000080--010(HEX) (SELRD(0), CMRDUN).
- C.8 CPU Status Word 2.
  - a. Report word 2; error if not 000700FF30--0(HEX).
- C.9 Select Status bit 1.
  - a. Set bit 32 in processor PTM register.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - d. Clear bit 32 in processor PTM register.
  - e. Read RF8, report data; error if not C3--C3000000(HEX).
- C.10 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0000080--010(HEX) (SELRD(0), CMRDUN).
- C.11 CPU Status Word 2.
  - a. Report word 2; error if not 000800FF30--0(HEX).
- C.12 Select status bit 2.
  - a. Set bit 31 in processor PTM register.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand 1, read word from RMA, wait for CPU halted.
  - d. Clear bit 31 in processor PTM register.
  - e. Read RF8, report data; error if not C3--C3000000(HEX).
- C.13 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0000080--010(HEX) (SELRD(0), CMRDUN).
- C.14 CPU Status Word 2.
  - a. Report word 2; error if not 000500FF30--0(HEX).
- C.15 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

60000285 A

LMT3/P - 38

## SECTION 2 - PRELIMINARY CACHE CHECKS

Verify the operation of various circuits associated with cache memory, preliminary to the testing of cache sets.

Subsection 0 - AC Adrs Bits/IF Adrs Bits through Adrs Mux 1 and 2 to Cache Sets 0 and 1.

Subsection 1 - AC Adrs Bits/IF Adrs Bits through Adrs Mux 1 and 2 to Cache Sets 2 and 3 if available.

### Method:

Micrands used in this section, other than Instruction Fetch (IF) sequences, all use the SVA option, eliminating the need for Segment Map initialization. The IF process is described earlier in this document under Special Procedures.

FAKECM mode is used exclusively in this section to inhibit Page Map and CM usage.

### Section 2, Subsection 0

This subsection tests AC/IF ADRS BITS (ACADDR/IFADDR) through Adrs Mux 1 and 2 (AMUX) to Cache Sets 0 and 1.

Test the 48-bit ACADDR and IFADDR address entries to and through the Address Mux (AMUX). ACADDR from Address Control and IFADDR from Instruction Fetch are used in this subsection to generate Cache Tag data.

### Hardware Tested:

- AC/IF ADRS BITS (ACADDR/IFADDR) 16 through 47.
- AC/IF ADRS BITS (ACADDR/IFADDR) 48 through 63.
- Cache Tag for Set 0.
- Cache Tag for Set 1.
- Cache Data for Set 0.
- Cache Data for Set 1.

### Method:

AMUX entries ACADDR and IFADDR are compared by using identical addresses to read the same cache location. The same read data is expected. There are two test series used in each pass through the conditions. The first series uses Cache Set 0, the second uses Set 1. In this way Cache Set data can also be compared. There are five passes, each using a different SVA pattern, to complete the subsection. All conditions use FAKECM mode.

60000285 A

LMT3/P - 39

Conditions 1 through 4 use Cache Set 0. Condition 1 selects Set 0 and allocates block 0 using a Tag defined by the current SVA pattern. Condition 2 writes a data pattern to word 0 of the allocated block. Condition 3 reads the data from word 0 and transfers it to the Register File. The address used comes from Address Control (ACADDR). The data read is expected to be the same as the data written. Condition 4 also reads data from word 0 using an address from Instruction Fetch (IFADDR). This data cannot be directly transferred to the Register File. It is used to address Control Store (CS) which then transfers a known constant to the Register File. See item 4 in Special Procedures.

Conditions 5 through 8 correspond to conditions 1 through 4, except Cache Set 1 is used.

If an error should occur in Conditions 3 and 4 and the data read equals zero, (Cache miss) suspect the Cache Tag board for Set 0. Suspect Set 1 if Conditions 7 and 8 read zeros.

If the result data displayed in condition 3 shows bits dropped or picked, suspect the Cache Data board for Set 0, Set 1 if Condition 7.

If Conditions 4 and 8 fail, suspect the AMUX boards or the possibility of bad addresses from either source.

The parity status received in conditions 4 and 8 will be masked to eliminate the PFS82(20-23) and PFS83(8-11) bits. The masking is performed because IF pre-fetching extra locations of cache may, under certain circumstances (such as a power-up of the hardware), include parity errors.

Test Patterns (hexadecimal):

| Pass | SVA            | PVA       | P Descriptor        |
|------|----------------|-----------|---------------------|
| 1    | 0000 0000 0000 | 0000 0000 | 0000 0000 0000 0000 |
| 2    | FFFF 7FFF E000 | 7FFF E000 | 0000 FFFF 0000 0000 |
| 3    | 5555 5555 4000 | 5555 4000 | 0000 5555 0000 0000 |
| 4    | AAAA 2AAA A000 | 2AAA A000 | 0000 AAAA 0000 0000 |
| 5    | EAEA 8AEA E000 | 8AEA E000 | 0000 EAEA 0000 0000 |

| Micrands Used: | Number | Description                 |
|----------------|--------|-----------------------------|
|                | C      | FAKECM Allocate Cache.      |
|                | D      | Read Word from SVA.         |
|                | E      | Write Word to SVA.          |
|                | F      | Purge Cache All.            |
|                | 10     | Set P Descriptor register.  |
|                | 13     | Enter P Register; start IF. |

Initialization:

- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Set Pass Count to 1.

60000285 B

LMT3/P - 40

Conditions:

- C.0 Set P Descriptor register and report pass count (1 through 5).
- Set RF11 to 0--0; segment number.
  - Set CM(0) to current P Descriptor value.
  - Start processor at micrand 10, set P Descriptor, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report pass count; informational.
- C.1 Purge cache, allocate current SVA to Set 0, report SVA.
- Select Cache Set 0.
  - Start processor at micrand F, purge all of cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to current SVA.
  - Select FAKECM mode.
  - Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.2 Write cache, report write data.
- Set RFA to F0--F0(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.3 Read cache using ACADDR, report read data.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not F0--F0.
- C.4 Read Cache Set 0 using IFADDR, report CS word.
- Set RF10 to current PVA.
  - Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand 13, enter P register (start IF), wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. Remove PFS82(20-23) and PFS83(8-11) from the actual status received before testing for errors.
  - Read RF14, report data; error if not A5--A5F0F0(HEX).
- C.5 Allocate current SVA to Set 1, report SVA.
- Deselect Cache Set 0.
  - Select Cache Set 1.
  - Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.

60000285 B

LMT3/P - 41



# Conditions:

- C.6 Write cache, report write data.
  - a. Set RFA to F1--F1(HEX); write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report write data; informational.
- C.7 Read Cache Using ACADDR, Report Read Data.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not F1--F1.
- C.8 Read Cache Set 1 Using IFADDR, Report CS Word.
  - a. Set RF14 to DF--DF(HEX); filler.
  - b. Start processor at micrand 13, enter P register, start IF, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0. Remove PFS82(20-23) and PFS83(8-11) from the actual status received before testing for errors.
  - d. Read RF14, report data; error if not A5--A5F1F1(HEX).
- C.9 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
  - b. Deselect FAKECM mode.
- C.11 First Failure Capture Information.
  - a. Report PFS87 bits 18-23 in bits 58-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit subsection if five passes complete; else get next SVA and return to C.0.

## Section 2, Subsection 1

This subsection tests AC/IF ADRS BITS (ACADDR/IFADDR) through Adrs Mux 1 and 2 (AMUX) to Cache Sets 2 and 3.

Test the 48-bit ACADDR and IFADDR ADDRESS BITS to and through the Address Mux (AMUX). ACADDR from Address Control and IFADDR from Instruction Fetch are used in this subsection to generate Cache Tag data.

## Hardware Tested:

- AC/IF ADRS BITS (ACADDR/IFADDR) 18 through 47.
- AC/IF ADRS BITS (ACADDR/IFADDR) 48 through 63.
- Cache Tag for Set 2.
- Cache Tag for Set 3.
- Cache Data for Set 2.
- Cache Data for Set 3.

## Method:

AMUX entries ACADDR and IFADDR are compared by using identical addresses to read the same cache location. The same read data is expected. There are two test series used in each pass through the conditions. The first series uses Cache Set 2, the second uses Set 3. In this way Cache Set data can also be compared. There are five passes, each using a different SVA pattern, to complete the subsection. All conditions use FAKECM mode.

Conditions 1 through 4 use Cache Set 2. Condition 1 selects Set 2 and allocates block 0 using a Tag defined by the current SVA pattern. Condition 2 writes a data pattern to word 0 of the allocated block. Condition 3 reads the data from word 0 and transfers it to the Register File. The address used comes from Address Control (ACADDR). The data read is expected to be the same as the data written. Condition 4 also reads data from word 0 using an address from Instruction Fetch (IFADDR). This data cannot be directly transferred to the Register File. It is used to address Control Store (CS) which then transfers a known constant to the Register File. See item 4 in Special Procedures.

Conditions 5 through 8 correspond to conditions 1 through 4, except Cache Set 3 is used.

If an error should occur in Conditions 3 and 4 and the data read equals zero, (Cache miss) suspect the Cache Tag board for Set 2. Suspect Set 3 if Conditions 7 and 8 read zeros.

If the result data displayed in condition 3 shows bits dropped or picked, suspect the Cache Data board for Set 2, Set 3 if Condition 7.

If Conditions 4 and 8 fail, suspect the AMUX boards or the possibility of bad addresses from either source.

The parity status received in conditions 4 and 8 will be masked to eliminate the PFS82(20-23) and PFS83(8-11) bits. The masking is performed because IF pre-fetching extra locations of cache may, under certain circumstances (such as a power-up of the hardware), include parity errors.

## Test Patterns (hexadecimal):

| Pass | SVA            | PVA       | P Descriptor        |
|------|----------------|-----------|---------------------|
| 1    | 0000 0000 0000 | 0000 0000 | 0000 0000 0000 0000 |
| 2    | FFFF 7FFF E000 | 7FFF E000 | 0000 FFFF 0000 0000 |
| 3    | 5555 5555 4000 | 5555 4000 | 0000 5555 0000 0000 |
| 4    | AAAA 2AAA A000 | 2AAA A000 | 0000 AAAA 0000 0000 |
| 5    | EAEA 8AEA E000 | 8AEA E000 | 0000 EAEA 0000 0000 |

# Micrands Used:

| Number | Description                  |
|--------|------------------------------|
| C      | FAKECM Allocate Cache.       |
| D      | Read Word from SVA.          |
| E      | Write Word to SVA.           |
| F      | Purge Cache All.             |
| 10     | Set P Descriptor Register.   |
| 13     | Enter P Register (Start IF). |

## Initialization:

- If Cache Sets 2 and 3 are not available, exit subsection.
- Set CM(0 through FFF(HEX) to C3--C30xxx(HEX); xxx = 0 through FFF(HEX).
- Set pass count to 1.

## Conditions:

- C.0 Set P Descriptor register and report pass count (1 through 5).
- Set RF11 to 0--0; segment number.
  - Set CM(0) to current P Descriptor value.
  - Start processor at micrand 10, set P Descriptor, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report pass count; informational.
- C.1 Purge cache, allocate current SVA to Set 0, report SVA.
- Select Cache Set 2.
  - Start processor at micrand F, purge all of cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to current SVA.
  - Select FAKECM mode.
  - Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.2 Write cache, report write data.
- Set RFA to F2--F2(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.

60000285 A

LMT3/P - 44

# Conditions:

- C.3 Read cache using ACADDR, report read data.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not F2--F2(HEX).
- C.4 Read Cache Set 2 using IFADDR, report CS word.
- Set RF10 to current PVA.
  - Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand 13, enter P register start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. Remove PFS82(20-23) and PFS83(8-11) from the actual status received before testing for errors.
  - Read RF14, report data; error if not A5--A5F2F2(HEX).
- C.5 Allocate current SVA to Set 1, report SVA.
- Select Cache Set 3.
  - Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.6 Write cache, report write data.
- Set RFA to F3--F3(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.7 Read cache using ACADDR, report read data.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not F3--F3(HEX).
- C.8 Read cache Set 3 Using IFADDR, Report CS Word.
- Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand 13, enter P register, start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. Remove PFS82(20-23) and PFS83(8-11) from the actual status received before testing for errors.
  - Read RF14, report data; error if not A5--A5F3F3(HEX).
- C.9 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
- Report word 2 if saved; error if reported.
  - Deselect FAKECM mode.

60000285 B

LMT3/P - 45

Conditions:

- C.11 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit subsection if five passes complete; else get next SVA and return to C.0.

SECTION 3 - CACHE DATA RAMS DATA INTEGRITY

Test the Cache Data RAMs ability to hold data in all cells.

Subsection 0 - Set 0 Data Test.

Subsection 1 - Set 1 Data Test.

Subsection 2 - Set 2 Data Test if available.

Subsection 3 - Set 3 Data Test if available.

Method:

The data test consists of allocating a Cache block, then writing and reading each one of the four words in that block with the four different data test patterns. Cache is allocated one block at a time by incrementing the block address portion of the SVA. The remaining SVA is held zero to reduce the possibility of a tag error. Data patterns used are 0s, Fs, 5s, and As.

FAKECM mode is selected.

Section 3, Subsection 0

This subsection tests the Cache Set 0 data integrity.

Test the Cache Data RAMs (1 by 1024 chips) and their associated logic to ensure Set 0 data integrity.

Hardware Tested:

- Cache Data RAMs for Set 0.
- Cache Tag Logic for Set 0.

Method:

Refer to section description.

# Test Patterns (hexadecimal):

| Pass | SVA            |
|------|----------------|
| 1    | 0000 0000 0000 |
| 2    | 0000 0000 0020 |
| 3    | 0000 0000 0040 |
| 4    | 0000 0000 0060 |
| 5    | 0000 0000 0080 |
| 6    | 0000 0000 00A0 |
|      |                |
| v    | v              |
| FF   | 0000 0000 1FC0 |
| 100  | 0000 0000 1FE0 |

## Data Written and Read

```

0000 0000 0000 0000
FFFF FFFF FFFF FFFF
5555 5555 5555 5555
AAAA AAAA AAAA AAAA
EAEA EAEA EAEA EAEA
3333 3333 3333 3333
CCCC CCCC CCCC CCCC
00FF 00FF 00FF 00FF
FF00 FF00 FF00 FF00
55AA 55AA 55AA 55AA
AA55 AA55 AA55 AA55

```

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

## Initialization:

- Select Cache Set 0.
- Set initial SVA to 0--0.
- Select FAKECM mode.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 100(HEX)).
- Report pass count; informational.

## Conditions:

- C.1 Current SVA.
- Set RF2 to current SVA.
  - Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.2 Expected result, word 0.
- Set RFA to current data pattern.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current data pattern; informational.
- C.3 Actual result, word 0.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not as reported in C.2.
- C.4 Expected result, word 1.
- Set RF2 to current SVA plus 8.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current data pattern; informational.
- C.5 Actual result, word 1.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not as reported in C.4.
- C.6 Expected result, word 2.
- Set RF2 to current SVA plus 10(HEX).
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current data pattern; informational.
- C.7 Actual result, word 2.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not as reported in C.6.

# Conditions:

- C.8 Expected result, word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.
- C.9 Actual result, word 3.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

If no errors, and all patterns not used, advance pattern and return to C.2.

Exit subsection if 100(HEX) passes complete; else advance SVA (+20(HEX) and return to C.0.

Deselect FAKECM mode.

## Section 3, Subsection 1

This subsection tests Cache Set 1 data integrity.

Test the Cache Data RAMs (1 by 1024 chips) and their associated logic to ensure set 1 data integrity.

## Hardware Tested:

- Cache Data RAMs for Set 1.
- Cache Tag Logic for Set 1.

## Method:

Refer to section description.

60000285 A

LMT3/P - 50

# Test Patterns (hexadecimal):

| Pass | SVA            |
|------|----------------|
| 1    | 0000 0000 0000 |
| 2    | 0000 0000 0020 |
| 3    | 0000 0000 0040 |
| 4    | 0000 0000 0060 |
| 5    | 0000 0000 0080 |
| 6    | 0000 0000 00A0 |
|      |                |
| V    | V              |
| FF   | 0000 0000 1FC0 |
| 100  | 0000 0000 1FE0 |

## Data Written and Read

0000 0000 0000 0000  
FFFF FFFF FFFF FFFF  
5555 5555 5555 5555  
AAAA AAAA AAAA AAAA  
EAEA EAEA EAEA EAEA

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

## Initialization:

- Select Cache Set 1.
- Set initial SVA to 0--0.
- Select FAKECM mode.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.

60000285 A

LMT3/P - 51

Conditions:

- C.2 Expected result, word 0.  
a. Set RFA to current data pattern.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.
- C.3 Actual result, word 0.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.
- C.4 Expected result, word 1.  
a. Set RF2 to current SVA plus 8.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.
- C.5 Actual result, word 1.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected result, word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.
- C.7 Actual result, word 2.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.6.
- C.8 Expected result, word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.

Conditions:

- C.9 Actual result, word 3.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

If no errors, and all patterns not used, advance pattern and return to C.2.

Exit subsection if 100(HEX) passes complete; else advance SVA (+20(HEX) and return to C.0.

Deselect FAKECM mode.

Section 3, Subsection 2

This subsection tests Cache Set 2 data integrity.

Test the Cache Data RAMs (1 by 1024 chips) and their associated logic to ensure set 2 data integrity.

Hardware Tested:

- Cache Data RAMs for Set 2.
- Cache Tag Logic for Set 2.

Method:

Refer to section description.

# Test Patterns (hexadecimal):

| Pass | SVA            |
|------|----------------|
| 1    | 0000 0000 0000 |
| 2    | 0000 0000 0020 |
| 3    | 0000 0000 0040 |
| 4    | 0000 0000 0060 |
| 5    | 0000 0000 0080 |
| 6    | 0000 0000 00A0 |
|      |                |
| v    | v              |
| FF   | 0000 0000 1FC0 |
| 100  | 0000 0000 1FE0 |

## Data Written and Read

0000 0000 0000 0000  
 FFFF FFFF FFFF FFFF  
 5555 5555 5555 5555  
 AAAA AAAA AAAA AAAA  
 EAEA EAEA EAEA EAEA

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

## Initialization:

- Select Cache Set 2 if available; if not exit subsection.
- Set initial SVA to 0--0.
- Select FAKECM mode.
- Set pass count to 1.

C.0 Pass count (1 through 100(HEX).  
 a. Report pass count; informational.

C.1 Current SVA.  
 a. Set RF2 to current SVA.  
 b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Report current SVA; informational.

## Conditions:

- C.2 Expected result, word 0.  
 a. Set RFA to current data pattern.  
 b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Report current data pattern; informational.
- C.3 Actual result, word 0.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not as reported in C.2.
- C.4 Expected result, word 1.  
 a. Set RF2 to current SVA plus 8.  
 b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Report current data pattern; informational.
- C.5 Actual result, word 1.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected result, word 2.  
 a. Set RF2 to current SVA plus 10(HEX).  
 b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Report current data pattern; informational.
- C.7 Actual result, word 2.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not as reported in C.6.
- C.8 Expected result, word 3.  
 a. Set RF2 to current SVA plus 18(HEX).  
 b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Report current data pattern; informational.

Conditions:

- C.9 Actual result, word 3.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

If no errors and all patterns not used, advance pattern and return to C.2.

Exit subsection if 100(HEX) passes complete; else advance SVA (+20(HEX) and return to C.0.

Deselect FAKECM mode.

Section 3, Subsection 3

This subsection tests Cache Set 3 data integrity.

Test the Cache Data RAMs (1 by 1024 chips) and their associated logic to ensure set 3 data integrity.

Hardware Tested:

- Cache Data RAMs for Set 3.
- Cache Tag Logic for Set 3.

Method:

Refer to section description.

Test Patterns (hexadecimal):

| Pass | SVA            |
|------|----------------|
| 1    | 0000 0000 0000 |
| 2    | 0000 0000 0020 |
| 3    | 0000 0000 0040 |
| 4    | 0000 0000 0080 |
| 5    | 0000 0000 0080 |
| 6    | 0000 0000 00A0 |
|      |                |
| v    | v              |
| FF   | 0000 0000 1FC0 |
| 100  | 0000 0000 1FE0 |

Data Written and Read

0000 0000 0000 0000  
FFFF FFFF FFFF FFFF  
5555 5555 5555 5555  
AAAA AAAA AAAA AAAA  
EAEA EAEA EAEA EAEA

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

Initialization:

- Select Cache Set 3 if available; if not exit subsection.
- Set initial SVA to 0--0.
- Select FAKECM mode.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.



Conditions:

- C.2 Expected result, word 0.  
a. Set RFA to current data pattern.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.
- C.3 Actual result, word 0.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.
- C.4 Expected result, word 1.  
a. Set RF2 to current SVA plus 8.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.
- C.5 Actual result, word 1.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected result, word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.
- C.7 Actual result, word 2.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.6.
- C.8 Expected result, word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current data pattern; informational.

Conditions:

- C.9 Actual result, word 3.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

If no errors and all patterns not used, advance pattern and return to C.2.

Exit subsection if 100(HEX) passes complete, else advance SVA (+20(HEX) and return to C.0.

Deselect FAKECM mode.

#### SECTION 4 - CACHE TAG RAMS DATA INTEGRITY

Test the Cache Tag RAMs (1 by 256 chips) for data integrity. The 34 Tag bits, the parity bit and the valid bit are tested.

Subsection 0 - Set 0 Tag RAMs.

Subsection 1 - Set 1 Tag RAMs.

Subsection 2 - Set 2 Tag RAMs if available.

Subsection 3 - Set 3 Tag RAMs if available.

##### Method:

Each Cache Tag location is tested in sequence using SVAs composed of a 34-bit test pattern and the Cache address.

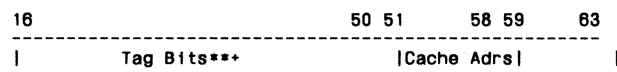


Figure LMT3/P-1. Typical SVA

\*\*\*Bit 32 is not included as part of the tag.

Three patterns are used: Fs, 5s, and As. Zeros were used in the same manner during the Cache Data RAM test.

A Cache Block is allocated, written and read with pattern one, written and read with pattern two, and written and read with pattern three. The block address is then incremented and the process repeated.

FAKECM mode is selected.

##### Section 4, Subsection 0

This subsection tests the Set 0 Cache Tag Array.

Test the Cache Tag RAMs (1 by 256 chips) and their associated logic to ensure all Tag locations in set 0 are usable.

##### Hardware Tested:

- Cache Tag RAMs for Set 0.

##### Method:

Refer to Section Method.

##### Test Patterns (hexadecimal):

| Pattern | SVA            |
|---------|----------------|
| 1       | FFFF 7FFF Ebb0 |
| 2       | 5555 2AAA Abb0 |
| 3       | AAAA 5555 4bb0 |

bb = The current cache block address. It varies from 00 through 1FE. It is shown as two full hexadecimal characters so the SVA pattern can be clearly seen.

Data written to word zero of each block is C3xx yyyy yyyy(Hex).

xx = Cache set selected.  
yy = Current SVA.

##### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

##### Initialization:

- Select Cache Set 0.
- Select FAKECM mode.
- Set initial Cache block address to zero.
- Set pass count to 1.

##### Conditions:

- C.0 Pass count (1 through 100(HEX)).
  - a. Report pass count; informational.
- C.1 Current pattern 1 SVA.
  - a. Set RF2 to FFFF 7FFF EBB0(HEX); bb = current block address.
  - b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report SVA; informational.

Conditions:

- C.2 Expected data pattern.  
a. Set RFA to C300 FFFF 7FFF EBB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.
- C.3 Actual data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.
- C.4 Current pattern 2 SVA.  
a. Set RF2 to 5555 2AAA ABB0(HEX); bb = current block address.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report SVA; informational.
- C.5 Expected data pattern.  
a. Set RFA to C300 5555 2AAA ABB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.
- C.6 Actual data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.5.
- C.7 Current pattern 3 SVA.  
a. Set RF2 to AAAA 5555 4BB0(HEX); bb = current block address.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report SVA; informational.
- C.8 Expected data pattern.  
a. Set RFA to C300 AAAA 5555 4BB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.

Conditions:

- C.9 Actual data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance block address +20(HEX) and return to C.0.

Deselect FAKECM mode.

Section 4, Subsection 1

This subsection tests the Set 1 Cache Tag Array.

Test the Cache Tag RAMs (1 by 256 chips) and their associated logic to ensure all Tag locations in set 1 are usable.

Hardware Tested:

- Cache Tag RAMs for Set 1.

Method:

See Section Method.

Test Patterns (hexadecimal):

| Pattern | SVA            |
|---------|----------------|
| 1       | FFFF 7FFF Ebb0 |
| 2       | 5555 2AAA Abb0 |
| 3       | AAAA 5555 4bb0 |

bb = The current cache block address. It varies from 00 through 1FE. It is shown as two full hexadecimal characters so the SVA pattern can be clearly seen.

Data written to word zero of each block is C3xx yyyy yyyy yyyy(HEX).

xx = Cache set selected.  
yy = Current SVA.

#### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

#### Initialization:

- Select Cache Set 1.
- Select FAKECM mode.
- Set initial Cache block address to zero.
- Set pass count to 1.

#### Conditions:

- C.0 Pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Current pattern 1 SVA.  
a. Set RF2 to FFFF 7FFF EBB0(HEX); bb = current block address.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report SVA; informational.
- C.2 Expected data pattern.  
a. Set RFA to C301 FFFF 7FFF EBB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.
- C.3 Actual data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.

#### Conditions:

- C.4 Current pattern 2 SVA.  
a. Set RF2 to 5555 2AAA ABB0(HEX); bb = current block address.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report SVA; informational.
- C.5 Expected data pattern.  
a. Set RFA to C301 5555 2AAA ABB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.
- C.6 Actual data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.5.
- C.7 Current pattern 3 SVA.  
a. Set RF2 to AAAA 5555 4BB0(HEX); bb = current block address.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report SVA; informational.
- C.8 Expected data pattern.  
a. Set RFA to C301 AAAA 5555 4BB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.
- C.9 Actual data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

#### Conditions:

Exit Subsection if 100(HEX) passes complete; else advance block address;  
+20(HEX), and return to C.0.

Deselect FAKECM mode.

#### Section 4, Subsection 2

This subsection tests the Set 2 Cache Tag Array.

Test the Cache Tag RAMs (1 by 256 chips) and their associated logic to ensure  
all Tag locations in set 2 are usable.

#### Hardware Tested:

- Cache Tag RAMs for Set 2.

#### Method:

Refer to Section Method.

#### Test Patterns (hexadecimal):

| Pattern | SVA            |
|---------|----------------|
| 1       | FFFF 7FFF Ebb0 |
| 2       | 5555 2AAA Abb0 |
| 3       | AAAA 5555 4bb0 |

bb = The current Cache block address. It varies from 00 through 1FE.  
It is shown as two full hexadecimal characters so the SVA pattern  
can be clearly seen.

Data written to word zero of each block is C3xx yyyy yyyy(HEX).

xx = Cache set selected.  
yy = Current SVA.

#### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

60000285 A

LMT3/P - 66

#### Initialization:

- Select Cache Set 2 if available, if not exit subsection.
- Select FAKECM mode.
- Set initial cache block address to zero.
- Set pass count to 1.

#### Conditions:

- C.0 Pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Current pattern 1 SVA.  
a. Set RF2 to FFFF 7FFF EBB0(HEX); bb = current block address.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU  
halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report SVA; informational.
- C.2 Expected data pattern.  
a. Set RFA to C302 FFFF 7FFF EBB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU  
halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.
- C.3 Actual data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU  
halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.
- C.4 Current pattern 2 SVA.  
a. Set RF2 to 5555 2AAA ABB0(HEX); bb = current block address.  
b. Start processor at micrand C, FAKECM allocate cache, wait for CPU  
halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report SVA; informational.
- C.5 Expected data pattern.  
a. Set RFA to C302 5555 2AAA ABB0(HEX); write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU  
halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report data; informational.

60000285 A

LMT3/P - 67

#### Conditions:

- C.6 Actual data.  
 a. Set RF6 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF6, report data; error if not as reported in C.5.
- C.7 Current pattern 3 SVA.  
 a. Set RF2 to AAAA 5555 4BB0(HEX); bb = current block address.  
 b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Report SVA; informational.
- C.8 Expected data pattern.  
 a. Set RFA to C302 AAAA 5555 4BB0(HEX); write data.  
 b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Report data; informational.
- C.9 Actual data.  
 a. Set RF6 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF6, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
 a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
 a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
 a. Report PFS87 bits 19-23 in bits 59-63; information only.  
 b. Report last PTL written in bits 34-47; information only.  
 c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance block address +20(HEX) and return to C.0.

Deselect FAKECM mode.

#### Section 4, Subsection 3

This subsection tests the Set 3 Cache Tag Array.

Test the Cache Tag RAMs (1 by 256 chips) and their associated logic to ensure all Tag locations in set 3 are usable.

60000285 A

LMT3/P - 68

#### Hardware Tested:

- Cache Tag RAMs for Set 3.

#### Method:

Refer to Section Method.

#### Test Patterns (hexadecimal):

| Pattern | SVA            |
|---------|----------------|
| 1       | FFFF 7FFF Ebb0 |
| 2       | 5555 2AAA Abb0 |
| 3       | AAAA 5555 4bb0 |

bb = The current Cache block address. It varies from 00 through 1FE. It is shown as two full hexadecimal characters so the SVA pattern can be clearly seen.

Data written to word zero of each block is C3xx yyyy yyyy yyyy(HEX).

xx = Cache set selected.  
 yy = Current SVA.

#### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

#### Initialization:

- Select Cache Set 3 if available; if not exit subsection.
- Select FAKECM mode.
- Set initial Cache block address to zero.
- Set pass count to 1.

#### Conditions:

- C.0 Pass count (1 through 100(HEX)).  
 a. Report pass count; informational.

60000285 A

LMT3/P - 69

Conditions:

- C.1 Current pattern 1 SVA.
  - a. Set RF2 to FFFF 7FFF EBB0(HEX); bb = current block address.
  - b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report SVA; informational.
- C.2 Expected data pattern.
  - a. Set RFA to C303 FFFF 7FFF EBB0(HEX); write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report data; informational.
- C.3 Actual data.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not as reported in C.2.
- C.4 Current pattern 2 SVA.
  - a. Set RF2 to 5555 2AAA ABB0(HEX); bb = current block address.
  - b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report SVA; informational.
- C.5 Expected data pattern.
  - a. Set RFA to C303 5555 2AAA ABB0(HEX); write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report data; informational.
- C.6 Actual data.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not as reported in C.5.
- C.7 Current pattern 3 SVA.
  - a. Set RF2 to AAAA 5555 4BB0(HEX); bb = current block address.
  - b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report SVA; informational.

Conditions:

- C.8 Expected data pattern.
    - a. Set RFA to C303 AAAA 5555 4BB0(HEX); write data.
    - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
    - c. Read CPU status, format and save if error; error if not 0.
    - d. Report data; informational.
  - C.9 Actual data.
    - a. Set RF8 to DF--DF(HEX); filler.
    - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
    - c. Read CPU status, format and save if error; error if not 0.
    - d. Read RF8, report data; error if not as reported in C.8.
  - C.10 CPU Status Word 1.
    - a. Report word 1 if saved; error if reported.
  - C.11 CPU Status Word 2.
    - a. Report word 2 if saved; error if reported.
  - C.12 First Failure Capture Information.
    - a. Report PFS87 bits 19-23 in bits 59-63; information only.
    - b. Report last PTL written in bits 34-47; information only.
    - c. Report last PTA written in bits 13-31; information only.
- Exit Subsection if 100(HEX) passes complete; else advance block address +20(HEX) and return to C.0.
- Deselect FAKECM mode.

SECTION 5 - CACHE TAG COMPARE NETWORK

Test the 34-bit Cache Tag Comparator logic in each available set. This network compares the Cache Tag In (from CAR) and the Cache Tag Out (from the Tag RAMs) to determine if a Cache match has occurred.

Subsection 0 - Set 0 Comparator.

Subsection 1 - Set 1 Comparator.

Subsection 2 - Set 2 Comparator; if available.

Subsection 3 - Set 3 Comparator; if available.

Method:

During the previous section many Tag match combinations were used and verified. This section tests the comparator for false match signals. Tag patterns of all zeros with a sliding one and all ones with a sliding zero, are used. The pattern is first tested to achieve a match. Then the test bit is cleared, or set, and used again to ensure a match does not occur.

FAKECM mode is selected and all allocations use cache block zero.

Section 5, Subsection 0

This subsection tests the Set 0 Cache Tag Comparator.

Test the Cache Set 0 Tag Comparator for detection of correct matches as well as false matches.

Hardware Tested:

- Set 0 Tag Comparator.

Method:

Refer to section description.

Test Patterns (hexadecimal):

| Pass | Positive<br>SVA | Complemented<br>SVA |
|------|-----------------|---------------------|
| 1    | 8000 0000 0000  | 7FFF 7FFF E000***   |
| 2    | 4000 0000 0000  | 8FFF 7FFF E000      |
| 3    | 2000 0000 0000  | DFFF 7FFF E000      |
|      |                 |                     |
| v    | v               | v                   |
| 10   | 0001 0000 0000  | FFFE 7FFF E000      |
| 11   | 0000 4000 0000  | FFFF 3FFF E000      |
|      |                 |                     |
| v    | v               | v                   |
| 21   | 0000 0000 4000  | FFFF 7FFF A000      |
| 22   | 0000 0000 2000  | FFFF 7FFF C000      |

\*\*\*Bit 32 of the SVA is always clear and is not part of the Cache Tag.

Data written to Cache = C3xx yyyy yyyy yyyy(HEX).

xx = 00 for Set 0.  
yy = Current SVA.

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

Initialization:

- Purge Cache all.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Cache Set 0.
- Select FAKECM mode.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 22(HEX)).
- a. Report pass count; informational.



Conditions:

- C.1 Current SVA, positive format.
  - a. Set RF2 to current positive SVA.
  - b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current SVA; informational.
- C.2 Expected data from hit.
  - a. Set RFA to current write pattern.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current write pattern; informational.
- C.3 Actual data from hit.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not as reported in C.2.
- C.4 Actual data from miss.
  - a. Set RF2 to 0--0 (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not 0.
- C.5 Current SVA, complement format.
  - a. Set RF2 to complement of current SVA.
  - b. Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current SVA; informational.
- C.6 Expected data from hit.
  - a. Set RFA to current write pattern.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current write pattern; informational.
- C.7 Actual data from hit.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not as reported in C.6.

Conditions:

- C.8 Actual data from miss.
  - a. Set RF2 to FFFF 7FFF E000(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not 0.
- C.9 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 22(HEX) passes complete; else advance operand pointer, and return to C.0.

Deselect FAKECM mode.

Section 5, Subsection 1

This subsection tests the Set 1 Cache Tag Comparator.

Test the Cache Set 1 Tag Comparator for detection of correct matches as well as false matches.

Hardware Tested:

- Set 1 Tag Comparator.

Method:

Refer to section description.

# Test Patterns (hexadecimal):

| Pass | Positive<br>SVA | Complemented<br>SVA |
|------|-----------------|---------------------|
| 1    | 8000 0000 0000  | 7FFF 7FFF E000***   |
| 2    | 4000 0000 0000  | BFFF 7FFF E000      |
| 3    | 2000 0000 0000  | DFFF 7FFF E000      |
|      |                 |                     |
| v    | v               | v                   |
| 10   | 0001 0000 0000  | FFFE 7FFF E000      |
| 11   | 0000 4000 0000  | FFFF 3FFF E000      |
|      |                 |                     |
| v    | v               | v                   |
| 21   | 0000 0000 4000  | FFFF 7FFF A000      |
| 22   | 0000 0000 2000  | FFFF 7FFF C000      |

\*\*\*Bit 32 of the SVA is always clear and not part of the cache tag.

Data written to Cache = C3xx yyyy yyyy yyyy(HEX).

xx = 01 for Set 1.  
yy = Current SVA.

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

## Initialization:

- Purge cache all.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero, if error, set failing condition number to -1(FE(HEX)).
- Select Cache Set 1.
- Select FAKECM mode.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 22(HEX)).
- Report pass count; informational.

## Conditions:

- C.1 Current SVA, positive format.
- Set RF2 to current positive SVA.
  - Start processor at micrand C, FAKECM allocate cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.2 Expected data from hit.
- Set RFA to current write pattern.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current write pattern; informational.
- C.3 Actual data from hit.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not as reported in C.2.
- C.4 Actual data from miss.
- Set RF2 to 0--0 (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0.
- C.5 Current SVA, complement format.
- Set RF2 to complement of current SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.6 Expected data from hit.
- Set RFA to current write pattern.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current write pattern; informational.
- C.7 Actual data from hit.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not as reported in C.6.

# Conditions:

- C.8 Actual data from miss.  
a. Set RF2 to FFFF 7FFF E000(HEX) (SVA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.
- C.9 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 22(HEX) passes complete; else advance operand pointer and return to C.0.

Deselect FAKECM mode.

## Section 5, Subsection 2

This subsection tests the Set 2 Cache Tag Comparator.

Test the Cache Set 2 Tag Comparator for detection of correct matches as well as false matches.

### Hardware Tested:

- Set 2 Tag Comparator.

### Method:

Refer to section description.

80000285 A

LMT3/P - 78

# Test Patterns (hexadecimal):

| Pass | Positive SVA   | Complemented SVA  |
|------|----------------|-------------------|
| 1    | 8000 0000 0000 | 7FFF 7FFF E000*** |
| 2    | 4000 0000 0000 | 8FFF 7FFF E000    |
| 3    | 2000 0000 0000 | DFFF 7FFF E000    |
|      |                |                   |
| v    | v              | v                 |
| 10   | 0001 0000 0000 | FFFE 7FFF E000    |
| 11   | 0000 4000 0000 | FFFF 3FFF E000    |
|      |                |                   |
| v    | v              | v                 |
| 21   | 0000 0000 4000 | FFFF 7FFF A000    |
| 22   | 0000 0000 2000 | FFFF 7FFF C000    |

\*\*\*Bit 32 of the SVA is always clear and not part of the Cache Tag.

### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

### Initialization:

- Select Cache Set 2 if available, if not exit subsection.
- Purge Cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error, set failing condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Set pass count to 1.

### Conditions:

- C.0 Pass count (1 through 22(HEX).  
a. Report pass count; informational.
- C.1 Current SVA, positive format.  
a. Set RF2 to current positive SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.

80000285 A

LMT3/P - 79

Conditions:

- C.2 Expected data from hit.  
a. Set RFA to current write pattern.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write pattern; informational.
- C.3 Actual data from hit.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.
- C.4 Actual data from miss.  
a. Set RF2 to 0--0 (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not 0.
- C.5 Current SVA, complement format.  
a. Set RF2 to complement of current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.6 Expected data from hit.  
a. Set RFA to current write pattern.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write pattern; informational.
- C.7 Actual data from hit.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.6.
- C.8 Actual data from miss.  
a. Set RF2 to FFFF 7FFF E000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not 0.

Conditions:

- C.9 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 22(HEX) passes complete; else advance operand pointer and return to C.0.

Deselect FAKECM mode.

Section 5, Subsection 3

This subsection tests the Set 3 Cache Tag Comparator.

Test the Cache Set 3 Tag Comparator for detection of correct matches as well as false matches.

Hardware Tested:

- Set 3 Tag Comparator.

Method:

Refer to section description.

Test Patterns (hexadecimal):

| Pass | Positive<br>SVA | Complemented<br>SVA |
|------|-----------------|---------------------|
| 1    | 8000 0000 0000  | 7FFF 7FFF E000***   |
| 2    | 4000 0000 0000  | BFFF 7FFF E000      |
| 3    | 2000 0000 0000  | DFFF 7FFF E000      |
|      |                 |                     |
| v    | v               | v                   |
| 10   | 0001 0000 0000  | FFFE 7FFF E000      |
| 11   | 0000 4000 0000  | FFFF 3FFF E000      |
|      |                 |                     |
| v    | v               | v                   |
| 21   | 0000 0000 4000  | FFFF 7FFF A000      |
| 22   | 0000 0000 2000  | FFFF 7FFF C000      |

\*\*\*Bit 32 of the SVA is always clear and not part of the Cache Tag.

Data written to Cache = C3xx yyyy yyyy (HEX).

xx = 03 for Set 3.  
yy = Current SVA.

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

Initialization:

- Select Cache Set 3 if available, if not exit subsection.
- Purge Cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error, set failing condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 22(HEX).  
a. Report pass count; informational.
- C.1 Current SVA, positive format.  
a. Set RF2 to current positive SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Expected data from hit.  
a. Set RFA to current write pattern.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write pattern; informational.
- C.3 Actual data from hit.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.

Conditions:

- C.4 Actual data from miss.  
a. Set RF2 to 0--0 (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not 0.
- C.5 Current SVA, complement format.  
a. Set RF2 to complement of current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.6 Expected data from hit.  
a. Set RFA to current write pattern.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write pattern; informational.
- C.7 Actual data from hit.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.6.
- C.8 Actual data from miss.  
a. Set RF2 to FFFF 7FFF E000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not 0.
- C.9 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 22(HEX) passes complete; else advance operand pointer and return to C.0.

Deselect FAKECM mode.

SECTION 6 - CACHE ADDRESSING, EXTERNAL TO RAM

Test Cache addressing to both the Cache Tag RAMs and Cache Data RAMs. This section is designed to test the registers feeding the address to the RAMs, not the address registers within the RAMs, although both areas should be suspected if a failure occurs.

Subsection 0 - Set 0 External Addressing.

Subsection 1 - Set 1 External Addressing.

Subsection 2 - Set 2 External Addressing; if available.

Subsection 3 - Set 3 External Addressing; if available.

Method:

There are four address patterns used in this section: zeros (0--0), ones (0--01FF8(HEX)), and alternate bit patterns (0--01550(HEX), 0--0AA8(HEX). All Tag addresses in the set being tested are allocated and all data words written. The four specified addresses are then accessed and the read data checked for the expected pattern.

FAKECM mode is selected.

Section 6, Subsection 0

This subsection tests the Set 0 Cache Address registers.

Hardware Tested:

- Set 0 Address registers.

Method:

Refer to section description.

Test Patterns (hexadecimal):

These test patterns are used during allocation/write loop.

| Pass | SVA  |      |      |
|------|------|------|------|
| 1    | 0000 | 0000 | 0000 |
|      | 0000 | 0008 | 0008 |
|      | 0000 | 0010 | 0010 |
|      | 0000 | 0018 | 0018 |
| 2    | 0000 | 0020 | 0020 |
|      | 0000 | 0028 | 0028 |
|      | 0000 | 0030 | 0030 |
|      | 0000 | 0038 | 0038 |
| 100  | 0000 | 1FC0 | 1FC0 |
|      | 0000 | 1FC8 | 1FC8 |
|      | 0000 | 1FD0 | 1FD0 |
|      | 0000 | 1FD8 | 1FD8 |
| 100  | 0000 | 1FE0 | 1FE0 |
|      | 0000 | 1FE8 | 1FE8 |
|      | 0000 | 1FF0 | 1FF0 |
|      | 0000 | 1FF8 | 1FF8 |

The write data format is: C3xx yyyy yyyy yyyy(HEX).

xx = 00 for Set 0.  
yy = Current SVA.

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

Initialization:

- Select Cache Set 0.
- Select FAKECM mode.
- Set current SVA to zero.

Conditions:

- C.0 Allocate write pass count (1 through 100(HEX).  
a. Increment and report pass count; informational.

Conditions:

- C.1 Allocate current SVA.  
a. Set RF2 to current SVA. If doing repeat iteration, use SVA last used.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write word 0.  
a. Set RFA to formatted write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report write data; informational.
- C.3 Write word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.4 Write word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.5 Write word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.

Continue to next condition if error or if 100(HEX) passes complete; else omit C.6 to C.11.

- C.6 Read from 0s address.  
a. Set RF2 to 0--0 (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C300 0000 0000 0000(HEX).

Conditions:

- C.7 Read from 1s address.  
a. Set RF2 to 0--01FF81FF8(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C300 0000 1FF8 1FF8(HEX).
- C.8 Read from As Address.  
a. Set RF2 to 0--0AA80AA8(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C300 0000 0AA8 0AA8(HEX).
- C.9 Read from 5s Address.  
a. Set RF2 to 0--015501550(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C300 0000 1550 1550(HEX).
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit subsection if 100(HEX) passes complete; else advance SVA and return to C.0.

Deselect FAKECM mode.

Section 6, Subsection 1

This subsection tests the Set 1 Cache Address Registers.

Hardware Tested:

- Set 1 Address Registers.

# Method:

Refer to section description.

## Test Patterns (hexadecimal):

These test patterns are used during the allocation/write loop.

| Pass | SVA                                                                  |
|------|----------------------------------------------------------------------|
| 1    | 0000 0000 0000<br>0000 0008 0008<br>0000 0010 0010<br>0000 0018 0018 |
| 2    | 0000 0020 0020<br>0000 0028 0028<br>0000 0030 0030<br>0000 0038 0038 |
| ↓    | ↓                                                                    |
| FF   | 0000 1FC0 1FC0<br>0000 1FC8 1FC8<br>0000 1FD0 1FD0<br>0000 1FD8 1FD8 |
| 100  | 0000 1FE0 1FE0<br>0000 1FE8 1FE8<br>0000 1FF0 1FF0<br>0000 1FF8 1FF8 |

The write data format is: C3xx yyyy yyyy yyyy(HEX).

xx = 01 for Set 1.  
yy = Current SVA.

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

## Initialization:

- Select Cache Set 1.
- Select FAKECM mode.
- Set current SVA to zero.

60000285 A

LMT3/P - 88

## Conditions:

- C.0 Allocate write pass count (1 through 100(HEX).  
a. Increment and report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA. If doing repeat iteration, use SVA last used.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write word 0.  
a. Set RFA to formatted write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report write data; informational.
- C.3 Write word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.4 Write word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.5 Write word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- Continue to next condition if error or if 100(HEX) passes complete; else omit C.6 to C.11.
- C.6 Read from 0's address.  
a. Set RF2 to 0--0 (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C301 0000 0000 0000(HEX).

60000285 A

LMT3/P - 89



# Conditions:

- C.7 Read from 1's address.
- Set RF2 to 0--01FF81FF8(HEX) (SVA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C301 0000 1FF8 1FF8(HEX).
- C.8 Read from A's address.
- Set RF2 to 0--0AA80AA8(HEX) (SVA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C301 0000 0AA8 0AA8(HEX).
- C.9 Read from 5's address.
- Set RF2 to 0--015501550(HEX) (SVA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C301 0000 1550 1550(HEX).
- C.10 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance SVA and return to C.0.

Deselect FAKECM mode.

## Section 6, Subsection 2

This subsection tests the Set 2 Cache Address Registers.

### Hardware Tested:

- Set 2 Address Registers.

60000285 A

LMT3/P - 90

# Method:

Refer to section description.

## Test Patterns (hexadecimal):

These test patterns are used during the allocation/write loop.

| Pass    | SVA                                                                                                                                               |
|---------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 1       | 0000 0000 0000<br>0000 0008 0008<br>0000 0010 0010<br>0000 0018 0018<br>0000 0020 0020<br>0000 0028 0028<br>0000 0030 0030<br>0000 0038 0038      |
| 2       | 0000 0020 0020<br>0000 0028 0028<br>0000 0030 0030<br>0000 0038 0038                                                                              |
| ↓<br>FF | ↓<br>0000 1FC0 1FC0<br>0000 1FC8 1FC8<br>0000 1FD0 1FD0<br>0000 1FD8 1FD8<br>0000 1FE0 1FE0<br>0000 1FE8 1FE8<br>0000 1FF0 1FF0<br>0000 1FF8 1FF8 |
| 100     |                                                                                                                                                   |

The write data format is: C3xx yyyy yyyy yyyy(HEX).

xx = 02 for Set 2.  
yy = Current SVA.

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

## Initialization:

- Select Cache Set 2 if available; if not exit subsection.
- Select FAKECM mode.
- Set current SVA to zero.

60000285 A

LMT3/P - 91

Conditions:

- C.0 Allocate write pass count (1 through 100(HEX)).  
a. Increment and report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA. If doing repeat iteration, use SVA last used.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write word 0.  
a. Set RFA to formatted write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report write data; informational.
- C.3 Write word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.4 Write word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.5 Write word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- Continue to next condition if error or if 100(HEX) passes complete; else omit C.6 to C.11.
- C.6 Read from 0's address.  
a. Set RF2 to 0--0 (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C302 0000 0000 0000(HEX).

60000285 A

LMT3/P - 92

Conditions:

- C.7 Read from 1's address.  
a. Set RF2 to 0--01FF81FF8(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C302 0000 1FF8 1FF8(HEX).
- C.8 Read from A's address.  
a. Set RF2 to 0--0AA80AA8(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C302 0000 0AA8 0AA8(HEX).
- C.9 Read from 5's address.  
a. Set RF2 to 0--015501550(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C302 0000 1550 1550(HEX).
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit subsection if 100(HEX) passes complete; else advance SVA and return to C.0.

Deselect FAKECM mode.

Section 8, Subsection 3

This subsection tests the Set 3 Cache Address Registers.

Hardware Tested:

- Set 3 Address Registers.

60000285 A

LMT3/P - 93

# Method:

Refer to section description.

# Test Patterns (hexadecimal):

These test patterns are used during the allocation/write loop.

| Pass | SVA            |
|------|----------------|
| 2    | 0000 0008 0008 |
|      | 0000 0010 0010 |
|      | 0000 0018 0018 |
|      | 0000 0020 0020 |
|      | 0000 0028 0028 |
|      | 0000 0030 0030 |
| 100  | 0000 0038 0038 |
|      | 0000 1FC0 1FC0 |
|      | 0000 1FC8 1FC8 |
|      | 0000 1FD0 1FD0 |
|      | 0000 1FD8 1FD8 |
|      | 0000 1FE0 1FE0 |
|      | 0000 1FE8 1FE8 |
|      | 0000 1FF0 1FF0 |
|      | 0000 1FF8 1FF8 |

The write data format is: C3xx yyyy yyyy(HEX).

xx = 03 for Set 3.  
yy = Current SVA.

# Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

# Initialization:

- Select Cache Set 3 if available; if not exit subsection.
- Select FAKECM mode.
- Set current SVA to zero.

80000285 A

LMT3/P - 94

# Conditions:

- C.0 Allocate write pass count (1 through 100(HEX).  
a. Increment and report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA. If doing repeat iteration, use SVA last used.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write word 0.  
a. Set RFA to formatted write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report write data; informational.
- C.3 Write word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.4 Write word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.5 Write word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.

Continue to next condition if error or if 100(HEX) passes complete; else omit C.6 to C.11.

- C.6 Read from 0's address.  
a. Set RF2 to 0--0 (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C303 0000 0000 0000(HEX).

80000285 A

LMT3/P - 95

Conditions:

- C.7 Read from 1's address.
- Set RF2 to 0--01FF81FF8(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C303 0000 1FF8 1FF8(HEX).
- C.8 Read from A's Address.
- Set RF2 to 0--0AA80AA8(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C303 0000 0AA8 0AA8(HEX).
- C.9 Read from 5's Address.
- Set RF2 to 0--015501550(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C303 0000 1550 1550(HEX).
- C.10 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

SECTION 7 - CACHE RAM ADDRESSING, TAG AND DATA

Test the individual Cache Data RAMs (1 by 1024) and the Cache Tag RAMs (1 by 256) for correct addressing.

Subsection 0 - Set 0 RAM Addressing.

Subsection 1 - Set 1 RAM Addressing.

Subsection 2 - Set 2 RAM Addressing; if available.

Subsection 3 - Set 3 RAM Addressing; if available.

Method:

A march procedure is used in this section to test the addressing of each individual bit. This is accomplished by:

- Writing all Cache words in the set being tested with a base pattern of A--A(HEX).
- Starting at word address zero, reading Cache and testing for A--A(HEX).
- Then writing a 54--54(HEX) pattern to the address just read.

Repeat this procedure (steps 2 and 3) until all words have been written with the 54--54(HEX) pattern. This completes the testing of all bits, including the parity bit, except for bit 7 of every byte. Bit 7 is now tested by repeating the procedure below, using a new write pattern.

- Read Cache address N, test for 54--54(HEX).
- Write a 01--01(HEX) pattern to address N.
- Increment N and repeat until all 1024 addresses have been tested.

FAKECM mode is selected.

Section 7, Subsection 0

This subsection tests Cache Set 0 RAM addressing.

Test Set Zero's Cache Tag and Cache Data RAMs for correct addressing.

Hardware Tested:

- Cache Data RAMs for Set 0.
- Cache Tag RAMs for Set 0.

Method:

Refer to section description.

Test Patterns (hexadecimal):

Base Pattern = AA--AA.  
First Write Pattern = 54--54.  
Second Write Pattern = 01--01.

SVAs from 0000 0000 0000 to 0000 0000 1FF8(HEX).

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

Initialization:

- Select Cache Set 0
- Select FAKECM mode.
- Set RFA to AA--AA(HEX); write data.
- Set current SVA to zero.

Conditions:

- C.0 Current SVA Allocated.
- a. Set RF2 to current SVA.
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current SVA; informational.
- C.1 Write data; all 4 words.
- a. Start processor at micrand E, write word to SVA; write word 0, wait for CPU halted.
  - b. Read CPU status, format and save if error; error if not 0.
  - c. Set RF2 to current SVA plus 8.
  - d. Start processor at micrand E, write word to SVA; write word 1, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Set RF2 to current SVA plus 10(HEX).
  - g. Start processor at micrand E, write word to SVA; write word 2, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Set RF2 to current SVA plus 18(HEX).
  - j. Start processor at micrand E, write word to SVA; write word 3, wait for CPU halted.
  - k. Read CPU status, format and save if error; error if not 0.
  - l. Report AA--AA(HEX); write data; informational.

60000285 A

LMT3/P - 98

Conditions:

Continue to next condition if error, SSM, or if all locations allocated and written; else advance SVA and return to C.0.

Clear SVA.

Set RFA to 54--54(HEX); write pattern.

- C.2 Current SVA for Test Part 1.
- a. Set RF2 to current SVA.
  - b. Report current SVA; informational.
- C.3 Read data for Test Part 1.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not AA--AA(HEX).
- C.4 Write data for Test Part 1.
- a. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - b. Read CPU status, format and save if error; error if not 0.
  - c. Report 54--54(HEX); informational.

Continue to next condition if error, SSM, or if all Cache words (1024) tested, else advance SVA to next word and return to C.2.

Clear SVA.

Set RFA to 01--01(HEX); write pattern.

- C.5 Current SVA for Test Part 2.
- a. Set RF2 to current SVA.
  - b. Report current SVA; informational.
- C.6 Read data for Test Part 2.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not 54--54(HEX).
- C.7 Write data for Test Part 2.
- a. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - b. Read CPU status, format and save if error; error if not 0.
  - c. Report 01--01(HEX); informational.
- C.8 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.

60000285 A

LMT3/P - 99

C.9 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if all Cache words (400(HEX) tested; else advance SVA, if Scope Mode not set, and return to C.5.

Deselect FAKECM mode.

#### Section 7, Subsection 1

This subsection tests Cache Set 1 RAM addressing.

Test Set One's Cache Tag and Cache Data RAMs for correct addressing.

#### Hardware Tested:

- Cache Data RAMs for Set 1.
- Cache Tag RAMs for Set 1.

#### Method:

Refer to section description.

Test Patterns (hexadecimal):

Base Pattern = AA--AA.  
First Write Pattern = 54--54.  
Second Write Pattern = 01--01.

SVAs from 0000 0000 0000 to 0000 0000 1FF8(HEX).

#### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

#### Initialization:

- Select Cache Set 1
- Select FAKECM mode.
- Set RFA to AA--AA(HEX); write data.
- Set current SVA to zero.

#### Conditions:

- C.0 Current SVA Allocated.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.1 Write data; all 4 words.  
a. Start processor at micrand E, write word to SVA; write word 0, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set RF2 to current SVA plus 8.  
d. Start processor at micrand E, write word to SVA; write word 1, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Set RF2 to current SVA plus 10(HEX).  
g. Start processor at micrand E, write word to SVA; write word 2, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Set RF2 to current SVA plus 18(HEX).  
j. Start processor at micrand E, write word to SVA; write word 3, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Report AA--AA(HEX); write data; informational.

Continue to next condition if error, SSM, or if all locations allocated and written, else advance SVA and return to C.0.

Clear SVA.

Set RFA to 54--54(HEX); write pattern.

- C.2 Current SVA for Test Part 1.  
a. Set RF2 to current SVA.  
b. Report current SVA; informational.
- C.3 Read data for Test Part 1.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not AA--AA(HEX).

Conditions:

- C.4 Write data for Test Part 1.
- Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Report 54--54(HEX); informational.

Continue to next condition if error, SSM, or if all Cache words (1024) tested; else advance SVA to next word and return to C.2.

Clear SVA.

Set RFA to 01--01(HEX); write pattern.

- C.5 Current SVA for Test Part 2.
- Set RF2 to current SVA.
  - Report current SVA; informational.
- C.6 Read Data for Test Part 2.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 54--54(HEX).
- C.7 Write Data for Test Part 2.
- Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report 01--01(HEX); informational.
- C.8 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if all Cache words (400(HEX) tested; else advance SVA, if Scope Mode not set, and return to C.5.

Deselect FAKECM mode.

Section 7, Subsection 2

This subsection tests Cache Set 2 RAM addressing.

Test Set Two's Cache Tag and Cache Data RAMs for correct addressing.

Hardware Tested:

- Cache Data RAMs for Set 2.
- Cache Tag RAMs for Set 2.

Method:

Refer to section description.

Test Patterns (hexadecimal):

Base Pattern = AA--AA.  
First Write Pattern = 54--54.  
Second Write Pattern = 01--01.

SVAs from 0000 0000 0000 to 0000 0000 1FF8(HEX).

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

Initialization:

- Select Cache Set 2 if available; if not exit subsection.
- Select FAKECM mode.
- Set RFA to AA--AA(HEX); write data.
- Set current SVA to zero.

Conditions:

- C.0 Current SVA Allocated.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.1 Write data; all 4 words.  
a. Start processor at micrand E, write word to SVA; write word 0, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set RF2 to current SVA plus 8.  
d. Start processor at micrand E, write word to SVA; write word 1, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Set RF2 to current SVA plus 10(HEX).  
g. Start processor at micrand E, write word to SVA; write word 2, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Set RF2 to current SVA plus 18(HEX).  
j. Start processor at micrand E, write word to SVA; write word 3, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Report AA--AA(HEX); write data; informational.

Continue to next condition if error, SSM, or if all locations allocated and written, else advance SVA and return to C.0.

Clear SVA.

Set RFA to 54--54(HEX); write pattern.

- C.2 Current SVA for Test Part 1.  
a. Set RF2 to current SVA.  
b. Report current SVA; informational.
- C.3 Read data for Test Part 1.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not AA--AA(HEX).
- C.4 Write data for Test Part 1.  
a. Start processor at micrand E, write word to SVA, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Report 54--54(HEX); informational.

Continue to next condition if error, SSM, or if all Cache words (1024) tested; else advance SVA to next word and return to C.2.

Clear SVA.

60000285 A

LMT3/P - 104

Conditions:

Set RFA to 01--01(HEX); write pattern.

- C.5 Current SVA for Test Part 2.  
a. Set RF2 to current SVA.  
b. Report current SVA; informational.
- C.6 Read data for Test Part 2.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not 54--54(HEX).
- C.7 Write data for Test Part 2.  
a. Start processor at micrand E, write word to SVA, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Report 01--01(HEX); informational.
- C.8 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if all Cache words (400(HEX) tested; else advance SVA, if Scope Mode not set, and return to C.5.

Deselect FAKECM mode.

Section 7, Subsection 3

This subsection tests Cache Set 3 RAM addressing.

Test Set Three's Cache Tag and Cache Data RAMs for correct addressing.

Hardware Tested:

- Cache Data RAMs for Set 3.
- Cache Tag RAMs for Set 3.

Method:

Refer to section description.

60000285 A

LMT3/P - 105



Test Patterns (hexadecimal):

Base Pattern = AA--AA.

First Write Pattern = 54--54.

Second Write Pattern = 01--01.

SVAs from 0000 0000 0000 to 0000 0000 1FF8(HEX).

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

Initialization:

- Select Cache Set 3 if available; if not exit subsection.
- Select FAKECM mode.
- Set RFA to AA--AA(HEX); write data.
- Set current SVA to zero.

Conditions:

- C.0 Current SVA Allocated.
- Set RF2 to current SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.1 Write data; all 4 words.
- Start processor at micrand E, write word to SVA; write word 0, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to current SVA plus 8.
  - Start processor at micrand E, write word to SVA; write word 1, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to current SVA plus 10(HEX).
  - Start processor at micrand E, write word to SVA; write word 2, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to current SVA plus 18(HEX).
  - Start processor at micrand E, write word to SVA; write word 3, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report AA--AA(HEX); write data, informational.

60000285 A

LMT3/P - 106

Conditions:

Continue to next condition if error, SSM, or if all locations allocated and written; else advance SVA and return to C.0.

Clear SVA.

Set RFA to 54--54(HEX); write pattern.

- C.2 Current SVA for Test Part 1.
- Set RF2 to current SVA.
  - Report current SVA; informational.
- C.3 Read data for Test Part 1.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not AA--AA(HEX).
- C.4 Write data for Test Part 1.
- Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report 54--54(HEX); informational.

Continue to next condition if error, SSM, or if all Cache words (1024) tested; else advance SVA to next word and return to C.2.

Clear SVA.

Set RFA to 01--01(HEX); write pattern.

- C.5 Current SVA for Test Part 2.
- Set RF2 to current SVA.
  - Report current SVA; informational.
- C.6 Read data for Test Part 2.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 54--54(HEX).
- C.7 Write data for Test Part 2.
- Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report 01--01(HEX); informational.
- C.8 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.
- Report word 2 if saved; error if reported.

60000285 A

LMT3/P - 107

- C.10 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Conditions:

Exit Subsection if all Cache words (400(HEX) tested; else advance SVA, if Scope Mode not set, and return to C.5.

Deselect FAKECM mode.

## SECTION 8 - PURGE CACHE ALL

Test the Purge Cache All operation and the LM Counter.

Subsection 0 - Set 0 Purge All.

Subsection 1 - Set 1 Purge All.

Subsection 2 - Set 2 Purge All; if available.

Subsection 3 - Set 3 Purge All; if available.

Method:

Conditions 0 through 2 allocate and write word 0 to all Cache blocks in sequential order. A Purge Cache All micrand is issued which should clear the Valid bit and set all 34 Tag-bits to ones in all locations. A Cache read is now issued to word zero of every Cache block in sequential order. No hits should occur and all zeros should be returned.

An error could occur if the Valid bit does not clear, but because the Tag was set to all ones it appears to clear. To test for this condition a second read is issued to the same Tag address using a Tag of all ones; for example, an SVA of FFFF 7FFF Exx0, where xx is the Tag address bits 51 through 58). No hits should occur and all zeros should be returned.

FAKECM mode is selected.

Section 8, Subsection 0

This subsection tests the Purge Cache All operation in Set 0.

Hardware Tested:

- Purge All Decode/Control.
- Adrs Mux B/C Inhibit Bits.
- LM Counter.
- Purge Cache All signal to Set 0.

Method:

Refer to section description.

# Test Patterns (hexadecimal):

| Pass     | First Read SVAs | Second Read SVAs |
|----------|-----------------|------------------|
| 1        | 0000 0000 0000  | FFFF 7FFF E000   |
| 2        | 0000 0000 0020  | FFFF 7FFF E020   |
| 3        | 0000 0000 0040  | FFFF 7FFF E040   |
| ↓        | ↓               | ↓                |
| 100(HEX) | 0000 0000 1FE0  | FFFF 7FFF FFE0   |

Data written = C3xx yyyy yyyy yyyy(HEX).

xx = Number of set being tested.  
yy = Current SVA.

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

## Initialization:

- Select Cache Set 0.
- Select FAKECM mode.
- Set current SVA to zero.
- Set pass count to 1.

## Conditions:

- C.0 Allocate/write pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write data to word 0.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write data; informational.

## Conditions:

Continue to next condition if error, SSM, or if 100(HEX) pass complete; else advance SVA (+20(HEX) and return to C.0.

Clear SVA.

Set pass count to 1.

- C.3 Execute Purge.  
a. Start processor at micrand F, purge Cache all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Report 0s; informational.
- C.4 Pass count for read loop (1 through 100(HEX).  
a. Report pass count; informational.
- C.5 Read data.  
a. Set RF2 to current SVA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.
- C.6 Read data, all 1's tag.  
a. Set RF2 to all 1s tag plus current SVA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.
- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance SVA, if Scope Mode not set, and return to C.4.

Deselect FAKECM mode.

## Section 8, Subsection 1

This subsection tests the Purge Cache All operation in Set 1.

### Hardware Tested:

- Purge Cache All signal to Set 1.

### Method:

Refer to section description.

### Test Patterns (hexadecimal):

| Pass     | First Read SVAs | Second Read SVAs |
|----------|-----------------|------------------|
| 1        | 0000 0000 0000  | FFFF 7FFF E000   |
| 2        | 0000 0000 0020  | FFFF 7FFF E020   |
| 3        | 0000 0000 0040  | FFFF 7FFF E040   |
|          |                 |                  |
| v        | v               | v                |
| 100(HEX) | 0000 0000 1FE0  | FFFF 7FFF FFE0   |

Write data = C3xx yyyy yyyy yyyy(HEX).

- xx = Number of set being tested.
- y--y = Current SVA.

### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

### Initialization:

- Select Cache Set 1.
- Select FAKECM mode.
- Set current SVA to zero.
- Set pass count to 1.

### Conditions:

- C.0 Allocate/write pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write Data to Word 0.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write data; informational.

Continue to next condition if error, SSM, or if 100(HEX) pass complete; else advance SVA (+20(HEX)) and return to C.0.

Clear SVA.

Set pass count to 1.

- C.3 Execute purge.  
a. Start processor at micrand F, purge Cache all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Report 0s; informational.
- C.4 Pass count for read loop (1 through 100(HEX).  
a. Report pass count; informational.
- C.5 Read data.  
a. Set RF2 to current SVA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.
- C.6 Read data, all 1's tag.  
a. Set RF2 to all 1s tag plus current SVA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.
- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

# Conditions:

- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance SVA, if Scope Mode not set, and return to C.4.

Deselect FAKECM mode.

## Section 8, Subsection 2

This subsection tests the Purge Cache All operation in Set 2.

### Hardware Tested:

- Purge Cache All signal to Set 2.

### Method:

Refer to section description.

### Test Patterns (hexadecimal):

| Pass     | First Read SVAs | Second Read SVAs |
|----------|-----------------|------------------|
| 1        | 0000 0000 0000  | FFFF 7FFF E000   |
| 2        | 0000 0000 0020  | FFFF 7FFF E020   |
| 3        | 0000 0000 0040  | FFFF 7FFF E040   |
| ↓        | ↓               | ↓                |
| 100(HEX) | 0000 0000 1FE0  | FFFF 7FFF FFE0   |

Write data = C3xx yyyy yyyy yyyy(HEX).

xx = Number of set being tested.  
y--y = Current SVA.

# Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

## Initialization:

- Select Cache Set 2 if available, if not exit subsection.
- Select FAKECM mode.
- Set current SVA to zero.
- Set pass count to 1.

# Conditions:

- C.0 Allocate/write pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write data to word 0.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write data; informational.

Continue to next condition if error, SSM, or if 100(HEX) pass complete; else advance SVA (+20(HEX)) and return to C.0.

Clear SVA.

Set pass count to 1.

- C.3 Execute purge.  
a. Start processor at micrand F, purge Cache all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Report 0s; informational.
- C.4 Pass count for read loop (1 through 100(HEX).  
a. Report pass count; informational.

# Conditions:

- C.5 Read data.  
a. Set RF2 to current SVA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.
- C.6 Read data, all 1s tag.  
a. Set RF2 to all 1s tag plus current SVA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.
- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance SVA, if Scope Mode not set, and return to C.4.

Deselect FAKECM mode.

## Section 8, Subsection 3

This subsection tests the Purge Cache All operation in Set 3.

## Hardware Tested:

- Purge Cache All signal to Set 3.

## Method:

Refer to section description.

60000285 A

LMT3/P - 116

# Test Patterns (hexadecimal):

| Pass     | First Read SVAs | Second Read SVAs |
|----------|-----------------|------------------|
| 1        | 0000 0000 0000  | FFFF 7FFF E000   |
| 2        | 0000 0000 0020  | FFFF 7FFF E020   |
| 3        | 0000 0000 0040  | FFFF 7FFF E040   |
|          |                 |                  |
| v        | v               | v                |
| 100(HEX) | 0000 0000 1FE0  | FFFF 7FFF FFE0   |

Write data = C3xx yyyy yyyy yyyy(HEX).

xx = Number of set being tested.  
y--y = Current SVA.

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

## Initialization:

- Select Cache Set 3 if available, if not exit subsection.
- Select FAKECM mode.
- Set current SVA to zero.
- Set pass count to 1.

## Conditions:

- C.0 Allocate/write pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.
- C.2 Write data to word 0.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write data; informational.

Continue to next condition if error, SSM, or if 100(HEX) pass complete; else advance SVA (+20(HEX) and return to C.0.

60000285 A

LMT3/P - 117

Conditions:

Clear SVA.

Set pass count to 1.

C.3 Execute purge.

- a. Start processor at micrand F, purge Cache all, wait for CPU halted.
- b. Read CPU status, format and save if error; error if not 0.
- c. Report 0s; informational.

C.4 Pass count for read loop; 1 through 100(HEX).

- a. Report pass count; informational.

C.5 Read data.

- a. Set RF2 to current SVA.
- b. Set RF8 to DF--DF(HEX); filler.
- c. Start processor at micrand D, read word from SVA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF8, report data; error if not 0.

C.6 Read data, all 1's tag.

- a. Set RF2 to all 1s tag plus current SVA.
- b. Set RF8 to DF--DF(HEX); filler.
- c. Start processor at micrand D, read word from SVA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF8, report data; error if not 0.

C.7 CPU Status Word 1.

- a. Report word 1 if saved; error if reported.

C.8 CPU Status Word 2.

- a. Report word 2 if saved; error if reported.

C.9 First Failure Capture Information.

- a. Report PFS87 bits 19-23 in bits 59-63; information only.
- b. Report last PTL written in bits 34-47; information only.
- c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance SVA, if Scope Mode not set, and return to C.4.

Deselect FAKECM mode.

## SECTION 9 - CACHE USABLE SIGNAL FROM MAC

Test the Cache configuration bit to ensure it disables the Cache match and allocate logic.

Subsection 0 - Set 0 Cache Usable.

Subsection 1 - Set 1 Cache Usable.

Subsection 2 - Set 2 Cache Usable; if available.

Subsection 3 - Set 3 Cache Usable; if available.

Method:

Bits 57 through 60 in the processor Dependent Environment Control (DEC) register are used to enable/disable specific Cache sets. When a set or sets are disabled certain areas of Cache are affected. This section tests those areas.

The disable to the Cache match logic is tested by allocating a Cache block, writing, and then reading word 0 to ensure the match. The set is now disabled and the same read is tried again; this time no match should occur and zeros should be returned.

The disable allocate is tested by issuing an allocate to a Cache block with the set disabled. The set is then enabled and written to and read from. A miss should occur and zeros returned.

FAKECM mode is selected.

Section 9, Subsection 0

This subsection tests the Cache Set Zero enable/disable logic.

Hardware Tested:

- Set Usable Logic for Set 0.

Method:

Refer to section description.

Test Patterns:

Refer to conditions.

# Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

## Initialization:

- None.

## Conditions:

- C.0 Select Set 0 and Allocate Address 0.
  - a. Start processor at micrand F, purge Cache all, wait for CPU halted.
  - b. Read CPU status, format and save if error; error if not 0.
  - c. Select FAKECM mode.
  - d. Select Cache Set 0, disable all others.
  - e. Set RF2 to 0--0(HEX) (SVA).
  - f. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - g. Read CPU status, format and save if error; error if not 0.
  - h. Report SVA; informational.
- C.1 Write to address 0.
  - a. Set RFA to C3C3 0000 C3C3 0000(HEX); write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report write data; informational.
- C.2 Read from Set 0 when Enabled.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not C3C3 0000 C3C3 0000(HEX).
- C.3 Read from Set 0 when disabled.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Disable set 0.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not 0.

## Conditions:

- C.4 Allocate address 1, Set disabled.
  - a. Set RF2 to 0--020(HEX) (SVA).
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report SVA; informational.
- C.5 Enable Set 0 and write address 1.
  - a. Select Set 0.
  - b. Set RFA to C3C3 0000 C3C3 0001; write data.
  - c. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report write data; informational.
- C.6 Read from address 1.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not 0--0.
- C.7 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

## Section 9, Subsection 1

This subsection tests the Cache Set One enable/disable logic.

## Hardware Tested:

- Set Usable Logic for Set 1.

## Method:

Refer to section description.



# Test Patterns:

Refer to conditions.

# Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

# Initialization:

- None.

# Conditions:

- C.0 Select Set 1 and allocate address 0.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Select FAKECM mode.
  - Select Cache Set 1, disable all others.
  - Set RF2 to 0--0 (SVA).
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.1 Write to address 0.
- Set RFA to C3C3 0001 C3C3 0000(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.2 Read from Set 1 when enabled.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C3C3 0001 C3C3 0000(HEX).

# Conditions:

- C.3 Read from Set 1 when disabled.
- Set RF8 to DF--DF(HEX); filler.
  - Disable set 1.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0.
- C.4 Allocate address 1, Set disabled.
- Set RF2 to 0--020(HEX) (SVA).
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.5 Enable Set 1 and write address 1.
- Select Set 1.
  - Set RFA to C3C3 0001 C3C3 0001; write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.6 Read from address 1.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0--0.
- C.7 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

# Section 9, Subsection 2

This subsection tests the Cache Set Two enable/disable logic.

# Hardware Tested:

- Set Usable Logic for Set 2.

#### Method:

Refer to section description.

#### Test Patterns:

Refer to conditions.

#### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

#### Initialization:

- Continue if Cache Set 2 available; if not exit subsection.

#### Conditions:

- C.0 Select Set 2 and allocate address 0.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Select FAKECM mode.
  - Select Cache Set 2 only.
  - Set RF2 to 0--0 (SVA).
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.1 Write to address 0.
- Set RFA to C3C3 0002 C3C3 0000(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.2 Read from Set 2 when enabled.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3C3 0002 C3C3 0000(HEX).

#### Conditions:

- C.3 Read from Set 2 when disabled.
- Set RF6 to DF--DF(HEX); filler.
  - Disable set 2.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0.
- C.4 Allocate address 1, Set disabled.
- Set RF2 to 0--020(HEX) (SVA).
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.5 Enable Set 2 and write address 1.
- Select Set 2.
  - Set RFA to C3C3 0002 C3C3 0001; write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.6 Read from address 1.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0--0.
- C.7 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

#### Section 9, Subsection 3

This subsection tests the Cache Set Three enable/disable logic.

#### Hardware Tested:

- Set Usable Logic for Set 3.

Method:

Refer to section description.

Test Patterns:

Refer to conditions.

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

Initialization:

- Continue if Cache Set 3 available; if not exit subsection.

Conditions:

- C.0 Select Set 3 and allocate address 0.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Select FAKECM mode.
  - Select Cache Set 3 only.
  - Set RF2 to 0--0(HEX) (SVA).
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.1 Write to address 0.
- Set RFA to C3C3 0003 C3C3 0000(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.2 Read from Set 3 when enabled.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C3C3 0003 C3C3 0000(HEX).

60000285 A

LMT3/P - 126

Conditions:

- C.3 Read from Set 3 when disabled.
- Set RF8 to DF--DF(HEX); filler.
  - Disable set 3.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0.
- C.4 Allocate address 1, Set disabled.
- Set RF2 to 0--020(HEX) (SVA).
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.5 Enable Set 3 and write address 1.
- Select Set 3.
  - Set RFA to C3C3 0003 C3C3 0001; write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.6 Read from address 1.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0--0.
- C.7 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

60000285 A

LMT3/P - 127

SECTION 10 - CACHE PARITY CIRCUITS

Test the Cache Tag parity circuits and the CAR parity checker. The Cache parity circuits are common to all Cache sets except the Tag-Out parity checker which is duplicated within each set. Also tested is the Cache write data parity checker.

- Subsection 0 - Set 0 Parity Circuits.
- Subsection 1 - Set 1 Parity Circuits.
- Subsection 2 - Set 2 Parity Circuits; if available.
- Subsection 3 - Set 3 Parity Circuits; if available.
- Subsection 4 - Cache Write Data Parity Checker.

Method (Subsections 0 through 3):

The Cache Tag-In and Tag-Out parity checkers are tested by using counting patterns to cover every possible bit combination per checker segment. Both checkers are organized into 9-bit segments. Therefore, if a counting pattern from 0 to 1FF(HEX) is used and repeated at 9-bit intervals, all bit combinations are tested. A few special patterns are required to test the 9-bit segment link.

The Tag parity generator is tested as a result of testing the checkers.

The CAR parity checker is organized into 8-bit segments. Therefore, it too is completely tested.

All test patterns are submitted using FAKECM mode Allocate, Write and Read.

Section 10, Subsection 0

This subsection tests the common parity circuits associated with Cache and the Set 0 Tag-Out parity checker.

Hardware Tested:

- Tag Parity Gen. (16 through 47).
- Tag Parity Gen. (48 through 50).
- Tag-In Parity Checker.
- Tag-Out Parity Checker for Set 0.
- CAR Parity for bits 16 through 47.
- CAR Parity for bits 48 through 60.
- Tag File Adrs Par Check for Set 0.

Method:

Refer to section description.

60000285 A

LMT3/P - 128

Test Patterns (hexadecimal):

| Pass | SVA (9-bit counting pattern) |
|------|------------------------------|
| 1    | 2010 0804 0201               |
| 2    | 4020 1008 0402               |
| 3    | 6030 180C 0603               |
| ↓    |                              |
| ↓    |                              |
| 1FF  | DFF F7FB FDFE                |
| 200  | FFF FFFF FFFF                |

NOTE

All SVAs are masked to clear bit 32 and bits 61 through 63. They are shown here before masking to better illustrate the counting pattern.

Special Patterns to Complete Test (hexadecimal):

| Pass | SVA                             |
|------|---------------------------------|
| 201  | 0000 0800 0000 (bit 36)         |
| 202  | 0010 0000 0000 (bit 27)         |
| 203  | 0010 0800 0000 (bits 27 and 36) |
| 204  | 2000 0000 0000 (bit 18)         |
| 205  | 2010 0000 0000 (bits 18 and 27) |

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word from SVA.   |

Initialization:

- Select FAKECM mode.
- Select Cache Set 0.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 205(HEX)).
  - a. Report pass count; informational.

60000285 A

LMT3/P - 129

# Conditions:

- C.1 Allocate current SVA.
  - a. Set RF2 to current SVA.
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Report SVA; informational.
- C.2 Report CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.3 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.4 Write Cache, tests Tag-Out Parity.
  - a. Set RFA to C300 YYYY YYYY YYYYH where YY-Y=current SVA; write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Report write data; informational.
- C.5 Report CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.6 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.7 Read Cache to verify data.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read RF8, report data; error if not as reported in C.4.
- C.8 Report CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.9 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.10 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 205(HEX) passes complete; else form next SVA and return to C.0.

Deselect FAKECM mode.

## Section 10, Subsection 1

This subsection tests the common parity circuits associated with Cache and the Set 1 Tag-Out parity checker.

80000285 A

LMT3/P - 130

# Hardware Tested:

- Tag-Out Parity Checker for Set 1.
- Tag File Adrs Par Check for Set 1.

## Method:

Refer to section description.

## Test Patterns (hexadecimal):

| Pass | SVA (9-bit counting pattern) |
|------|------------------------------|
| 1    | 2010 0804 0201               |
| 2    | 4020 1008 0402               |
| 3    | 6030 180C 0603               |
| ↓    |                              |
| V    |                              |
| 1FF  | DFF F7FB FDFE                |
| 200  | FFFF FFFF FFFF               |

## NOTE

All SVAs are masked to clear bit 32 and bits 61 through 63. They are shown here before masking to better illustrate the counting pattern.

## Special Patterns to Complete Test (hexadecimal):

| Pass | SVA                             |
|------|---------------------------------|
| 201  | 0000 0800 0000 (bit 36)         |
| 202  | 0010 0000 0000 (bit 27)         |
| 203  | 0010 0800 0000 (bits 27 and 36) |
| 204  | 2000 0000 0000 (bit 18)         |
| 205  | 2010 0000 0000 (bits 18 and 27) |

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word from SVA.   |

## Initialization:

- Select FAKECM mode.
- Select Cache Set 1.
- Set pass count to 1.

80000285 A

LMT3/P - 131

Conditions:

- C.0 Pass count (1 through 205(HEX)).
  - a. Report pass count; informational.
- C.1 Allocate current SVA.
  - a. Set RF2 to current SVA.
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Report SVA; informational.
- C.2 Report CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.3 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.4 Write Cache, tests Tag-Out Parity.
  - a. Set RFA to C301 YYYY YYYY YYYY(HEX) where YY-Y=current SVA; write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Report write data; informational.
- C.5 Report CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.6 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.7 Read Cache to verify data.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read RF6, report data; error if not as reported in C.4.
- C.8 Report CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.9 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.10 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 205(HEX) passes complete; else form next SVA and return to C.0.

Deselect FAKECM mode.

Section 10, Subsection 2

This subsection tests the common parity circuits associated with Cache and the Set 2 Tag-Out parity checker.

Hardware Tested:

- Tag-Out Parity Checker for Set 2.
- Tag File Adrs Par Check for Set 2.

Method:

Refer to section description.

Test Patterns (hexadecimal):

| Pass | SVA (9-bit counting pattern) |
|------|------------------------------|
| 1    | 2010 0804 0201               |
| 2    | 4020 1008 0402               |
| 3    | 6030 180C 0603               |
| 1    |                              |
| V    |                              |
| 1FF  | DFF F7FB FDFE                |
| 200  | FFF FFFF FFFF                |

NOTE

All SVAs are masked to clear bit 32 and bits 61 through 63. They are shown here before masking to better illustrate the counting pattern.

Special Patterns to Complete Test (hexadecimal):

| Pass | SVA                             |
|------|---------------------------------|
| 201  | 0000 0800 0000 (bit 36)         |
| 202  | 0010 0000 0000 (bit 27)         |
| 203  | 0010 0800 0000 (bits 27 and 36) |
| 204  | 2000 0000 0000 (bit 18)         |
| 205  | 2010 0000 0000 (bits 18 and 27) |

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word from SVA.   |

Initialization:

- Select FAKECM mode
- Select Cache Set 2 if available; if not exit subsection.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 205(HEX).  
a. Report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Report SVA; informational.
- C.2 Report CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.3 Report CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.4 Write Cache, tests Tag-Out Parity.  
a. Set RFA to C302 YYYY YYYY YYYYH where YY-Y=current SVA; write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Report write data; informational.
- C.5 Report CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.6 Report CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.7 Read Cache to verify data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read RF8, report data; error if not as reported in C.4.
- C.8 Report CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.9 Report CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 205(HEX) passes complete; else form next SVA and return to C.0.

Deselect FAKECM mode.

60000285 A

LMT3/P - 134

Section 10, Subsection 3

This subsection tests the common parity circuits associated with Cache and the Set 3 Tag-Out parity checker.

Hardware Tested:

- Tag-Out Parity Checker for Set 3.
- Tag File Adrs Par Check for Set 3.

Method:

Refer to section description.

Test Patterns (hexadecimal):

| Pass | SVA (9-bit counting pattern) |
|------|------------------------------|
| 1    | 2010 0804 0201               |
| 2    | 4020 1008 0402               |
| 3    | 6030 180C 0603               |
| ↓    |                              |
| 1FF  | DFF F7FB FDFE                |
| 200  | FFF FFFF FFFF                |

NOTE

All SVAs are masked to clear bit 32 and bits 61 through 63. They are shown here before masking to better illustrate the counting pattern.

Special Patterns to Complete Test (hexadecimal):

| Pass | SVA                             |
|------|---------------------------------|
| 201  | 0000 0800 0000 (bit 36)         |
| 202  | 0010 0000 0000 (bit 27)         |
| 203  | 0010 0800 0000 (bits 27 and 36) |
| 204  | 2000 0000 0000 (bit 18)         |
| 205  | 2010 0000 0000 (bits 18 and 27) |

60000285 A

LMT3/P - 135

# Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word from SVA.   |

## Initialization:

- Select FAKECM mode.
- Select Cache Set 3 if available; if not exit subsection.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 205(HEX).  
a. Report pass count; informational.
- C.1 Allocate current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Report SVA; informational.
- C.2 Report CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.3 Report CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.4 Write Cache, Tests Tag-Out Parity.  
a. Set RFA to C303 YYYY YYYY YYYYH where YY=Y=current SVA; write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Report write data; informational.
- C.5 Report CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.6 Report CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.7 Read Cache to Verify Data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read RF8, report data; error if not as reported in C.4.

60000285 A

LMT3/P - 136

## Conditions:

- C.8 Report CPU Status Word 1.  
a. Read CPU status, format and report word 1; error if not 0.
- C.9 Report CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 205(HEX) passes complete; else form next SVA and return to C.0.

Deselect FAKECM mode.

## Section 10, Subsection 4

This subsection tests the 64-bit write data parity checker. This ensures no false parity errors are generated.

## Hardware Tested:

- Write Data Parity Checker.

## Method:

Data words are passed from the Register File through LM where they are checked for correct parity and written to CM(0). The 64-bit data is checked in 8-bit (one-byte) segments. Therefore, all 64-bit patterns are in reality 8-bit patterns repeated in each of the eight bytes. The pattern is a counting pattern, incremented from 0 to FF(HEX) in all bytes simultaneously.

## Test Patterns (hexadecimal):

| Pass | Write Pattern       |
|------|---------------------|
| 1    | 0000 0000 0000 0000 |
| 2    | 0101 0101 0101 0101 |
| 3    | 0202 0202 0202 0202 |
| 1    |                     |
| 100  | FFFF FFFF FFFF FFFF |

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

60000285 A

LMT3/P - 137



#### Initialization:

- Select FAKECM mode.
- Set current write pattern to 0.
- Set pass count to 1.

#### Conditions:

- C.0 Allocate Cache address 0.
- a. Set RF2 to 0--0 (SVA).
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report 0s; informational.
- C.1 Pass count (1 through 100(HEX)).
- a. Report pass count; informational.
- C.2 Expected Data, Write Cache.
- a. Set RFA to current pattern; write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report write data; informational
- C.3 Actual data, read Cache.
- a. Set RFB to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RFB, report data; error if not as reported in C.2.
- C.4 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.5 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.6 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if error or 100(HEX) passes complete; else form next write pattern and return to C.1.

Deselect FAKECM mode.

#### SECTION 11 - CACHE PARITY ERROR SIGNALS

Test the parity error signals from LM to MAC to ensure parity errors, when sensed, are correctly logged.

Subsection 0 - Cache Write Data Parity Signals.

Subsection 1 - Mark Data Parity Signals.

Subsection 2 - Cache Address Parity Signals.

Subsection 3 - CACHE ADDRESS REGISTER (CAR) PE, and Tag-In PE Signals.

Subsection 4 - Cache Hit Code Status Verification.

Subsection 5 - Do nothing, but keep as a spare.

Subsection 6 - Cache RAM Parity Signals.

#### Method:

Refer to subsection descriptions.

#### Section 11, Subsection 0

This subsection tests the parity error signals generated by the Cache Write Data Parity Checker. Ensure all 8 CHWDPE bits are logged in the PFS82 register.

#### Hardware Tested:

- Write Data Parity Checkers.
- PFS82 bits 8 through 15.

#### Method:

Wrong parity data is written to Cache and the parity status sampled. Three patterns are used, one to test all 8 status bits at once, and two to test for alternate status bits.

Errors should also occur and be recorded by Address Control for the A-Stream.

FAKECM mode is used.

#### Test Patterns:

Refer to Conditions.

# Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| E      | Write Word to SVA.     |

## Initialization:

- Select FAKECM mode.

## Conditions:

- C.0 Allocate Cache, report CPU Status Word 1.
  - a. Set RF2 to 0--0 (SVA).
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and report word 1; error if not 0.
- C.1 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.2 Test all bits, report write pattern.
  - a. Set MAC data output bit 18 in processor PTM register.
  - b. Set RFA to 01--01(HEX); write data.
  - c. Clear bit 18 in processor PTM Register.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Report write data; informational.
- C.3 CPU Status Word 1.
  - a. Read CPU Status, format and report word 1; error if not 00FF0080--0(HEX); CHWDPE (0 through 7) and CHCPPE.
- C.4 CPU Status Word 2.
  - a. Report word 2; error if not 0--0FF0000(HEX); ADC A-Stream (0 through 7).
- C.5 Test even bits, report write pattern.
  - a. Set MAC data output bit (bit 18) in processor PTM register.
  - b. Set RFA to 0100--0100(HEX); write data.
  - c. Clear bit 18 in processor PTM register.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Report write data; informational.
- C.6 CPU Status Word 1.
  - a. Read CPU Status, format and report word 1; error if not 00AA0080--0(HEX); CHWDPE (0,2,4,6) and CHCPPE.
- C.7 CPU Status Word 2.
  - a. Report word 2; error if not 00--0AA0000(HEX); ADC A-Stream (0,2,4,6).

60000285 A

LMT3/P - 140

# Conditions :

- C.8 Test odd bits, report write pattern.
  - a. Set MAC data output bit 18 in processor PTM register.
  - b. Set RFA to 0001--0001(HEX); write data.
  - c. Clear bit 18 in processor PTM register.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Report write data; informational.
- C.9 CPU Status Word 1.
  - a. Read CPU Status, format and report word 1; error if not 00550080--0(HEX); CHWDPE (1,3,5,7) and CHCPPE.
- C.10 CPU Status Word 2.
  - a. Report word 2; error if not 0--0550000(HEX); ADC A-Stream (1,3,5,7).
- C.11 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

## Deselect FAKECM Mode.

## Section 11, Subsection 1

This subsection tests the Mark Data parity error signals to verify that errors can be detected.

## Hardware Tested:

- Mark Data Parity Signals.
- PFS82 Bit 27.

## Method:

The Mark Data Parity Checker has been previously tested by diagnostic ACT3/P. ACT3/P used all combinations of Mark bits during the Byte Read and Byte Write tests. This subsection only tests the ability of the Parity Checker to detect and log an error in PFS82 bit 27 (LMRKPE).

A Mark Data Parity Error is forced by setting the complement parity bit for the Mark Parity Generator in the processor PTM register and issuing a write operation. The parity generator, in Address Control, generates wrong parity which should be sensed by the checker in LM.

Two write operations are used, a write word which uses all 8 mark bits, and a write byte using 7 mark bits. This tests both even and odd patterns.

FAKECM mode is used.

60000285 A

LMT3/P - 141

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 11     | Write Byte to SVA. |

#### Initialization:

- Select FAKECM mode.

#### Conditions:

- C.0 Force LMRKPE with even data.
- Set RF2 to 0--0 (SVA).
  - Set RFA to DF---DF(HEX); write data.
  - Set Complement Mark Parity bit 9, in processor PTM registers.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Clear bit 9 in processor PTM register.
  - Report write data; informational.
- C.1 CPU Status Word 1.
- Read CPU Status, format and report word 1; error if not 00000010--0(HEX) (LMRKPE).
- C.2 CPU Status Word 2.
- Report word 2; error if not 0.
- C.3 Force LMRKPE with odd data.
- Set RF12 to 0--06(HEX); byte count, seven bytes.
  - Set Complement Mark Parity bit 9 in processor PTM registers.
  - Start processor at micrand 11, write bytes to SVA, wait for CPU halted.
  - Clear bit 9 in processor PTM register.
  - Report write data; informational.
- C.4 CPU Status Word 1.
- Read CPU Status, format and report word 1; error if not 00000010--0(HEX) (LMRKPE).
- C.5 CPU Status Word 2.
- Report word 2; error if not 0.

60000285 A

LMT3/P - 142

#### Conditions:

- C.6 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

#### Section 11, Subsection 2

This subsection tests the Cache Address Parity Signals.

Test both, or all four if four sets are available, CADRPE parity error signals, one from each set.

#### Hardware Tested:

- ADDYPE PE Signal.
- TGADPE PE Signal.
- CADRPE Register.

#### Method:

Cache address parity errors are forced by setting SVAs containing wrong parity into the Register File. The SVAs are then used to access Cache. Each set is selected in turn and tested. The expected CADRPE Status for the set under test should be sensed.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word From SVA.    |
| E      | Write Word to SVA.     |

#### Initialization:

- Select FAKECM mode.
- Set Pass Count to 1.

60000285 A

LMT3/P - 143

Conditions:

- C.0 Pass count (1 or 2).
  - a. Report pass count; informational.
- C.1 Select and allocate Set 0 or 2.
  - a. Select set 0; pass 1 or set 2, if available; pass 2.
  - b. Set RF2 to 0--020(HEX) (SVA).
  - c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report set selected; informational.
- C.2 Write Cache, report CPU Status Word 1.
  - a. Set MAC data output bit 18, in processor PTM register.
  - b. Set RF2 to 0--020(HEX) (SVA).
  - c. Clear bit 18 in processor PTM register.
  - d. Set RFA to C30x0--020(HEX); write data, x = set selected.
  - e. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - f. Read CPU status, format and report word 1; error if not 010000050--0(HEX); CARPE(5) and AMUXPEs.
- C.3 Report CPU Status Word 2.
  - a. Report word 2; error if not 80000000010--0(HEX); CADRPE(0) if Set 0 or 20000000010--0(HEX); CADRPE(2) if Set 2, SVA PE also set.
- C.4 Test disable of Parity Status, report CPU Status Word 1.
  - a. Disable Set 0; pass 1 or set 2, if available; pass 2.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and report word 1; error if not 010000050--0(HEX); CARPE(5) and AMUXPEs.
- C.5 Report CPU Status Word 2.
  - a. Report word 2; error if not 0--01000000(HEX); SVA PE only.
- C.6 Select and allocate Set 1 or 3.
  - a. Select set 1; pass 1 set 3, if available; pass 2.
  - b. Set RF2 to 0--020(HEX) (SVA).
  - c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report set selected; informational.
- C.7 Write Cache, report CPU Status Word 1.
  - a. Set MAC data output bit 18, in processor PTM register.
  - b. Set RF2 to 0--020(HEX) (SVA).
  - c. Clear bit 18 in processor PTM register.
  - d. Set RFA to C30x0--020(HEX); write data, x = set selected.
  - e. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - f. Read CPU status, format and report word 1; error if not 010000050--0(HEX); CARPE(5) and AMUXPEs.

60000285 A

LMT3/P - 144

Conditions:

- C.8 Report CPU Status Word 2.
  - a. Report word 2; error if not 40000000010--0(HEX) (CADRPE(1) if Set 1 or 10000000010--0(HEX); CADRPE(3) if Set 3, SVA PE also set.
- C.9 Test disable of Parity Status, report CPU Status Word 1.
  - a. Disable Set 1; pass 1 or set 3, if available; pass 2.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and report word 1; error if not 010000050--0(HEX); CARPE(5) and AMUXPEs.
- C.10 Report CPU Status Word 2.
  - a. Report word 2; error if not 0--01000000(HEX); SVA PE only.
- C.11 CPU Status Word 1 From C.1 or C.6 if Error.
  - a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2 From C.1 or C.6 if Error.
  - a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if only two Cache sets configured or if two passes complete, else return to C.0.

Deselect FAKECM mode.

Section 11, Subsection 3

This subsection tests the ability of the parity checkers to detect and transmit the appropriate error signals [CACHE ADDRESS REGISTER PE (CARPE)] to MAC for recording.

Hardware Tested:

- Tag-In PE Signal (CHTGPE).
- Tag-Out PE Signal (CRAMPE).
- CAR Parity Signals (CARPE).

Method:

As in the previous subsection, SVAs containing wrong parity in selected bytes are used to test the various parity signals.

60000285 A

LMT3/P - 145

The SVA is 48 bits or six bytes long. The six bytes, bytes two through seven, are individually set with wrong parity and tested. For example, conditions 0 and 1 are used to test byte seven. The MAC data output bit in the processor PTM register is used to force parity on MAC output data to zeros. The byte seven test SVA (0--08(HEX)) is set into the Register File and used to write Cache. A CARPE(8) should occur, the CHTGPE signal should not be generated during the byte seven test, but should occur on the remaining five tests. A Purge Cache micrand is issued prior to the word store micrand. This ensures that the CARPE Parity Error Signals, which are disabled during purges, are re-enabled during the write.

FAKECM mode is used.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description                     |
|--------|---------------------------------|
| 5      | Clear Residue Parity Errors.    |
| C      | FAKECM Allocate Cache.          |
| 27     | Purge Cache, Write Word to SVA. |

Initialization:

- Select FAKECM mode.

Conditions:

- C.0 Test byte seven, use Set zero, report CPU Status Word one.
- a. Select Cache Set 0.
  - b. Set RF2 to 0--08(HEX) (SVA).
  - c. Start CPU at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Set MAC data output bit 18, in processor PTM register.
  - f. Set RF2 to 0--08(HEX); SVA with wrong parity.
  - g. Clear bit 18 in processor PTM register.
  - h. Set RFA to C300--08(HEX); write data.
  - i. Start processor at micrand 27, purge Cache, write word to SVA, wait for CPU halted.
  - j. Read CPU status, format and report word 1; error if not 010--0 (HEX); CARPE(5).
- C.1 CPU Status Word 2.
- a. Report word 2; error if not 80--01000000(HEX); CADRPE(0) and SVAPE.

80000285 A

LMT3/P - 148

Conditions:

- C.2 Test byte six, use Set zero, report CPU Status Word one.
- a. Set RF2 to 0--08000(HEX) (SVA).
  - b. Start CPU at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Set MAC data output bit 18, in processor PTM register.
  - e. Set RF2 to 0--08000(HEX); SVA with wrong parity.
  - f. Clear bit 18 in processor PTM register.
  - g. Set RFA to C300--08000(HEX); write data.
  - h. Start processor at micrand 27, purge Cache, write word to SVA, wait for CPU halted.
  - i. Read CPU status, format and report word 1; error if not 020--0 (HEX); CARPE(4).
- C.3 CPU Status Word 2.
- a. Report word 2; error if not 80--02000000(HEX); CADRPE(0) and SVAPE.
- C.4 Test byte five, use Set one, report CPU Status Word one.
- a. Select Cache Set 1.
  - b. Set RF2 to 0--010000(HEX) (SVA).
  - c. Start CPU at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Set MAC data output bit 18, in processor PTM register.
  - f. Set RF2 to 0--010000(HEX); SVA with wrong parity.
  - g. Clear bit 18 in processor PTM register.
  - h. Set RFA to C301--010000(HEX); write data.
  - i. Start processor at micrand 27, purge Cache, write word to SVA, wait for CPU halted.
  - j. Read CPU status, format and report word 1; error if not 040--0 (HEX); CARPE(3).
- C.5 CPU Status Word 2.
- a. Report word 2; error if not 0--04000000(HEX) (SVAPE).
- C.6 Test byte four, use Set two or zero report CPU Status Word one.
- a. Select Cache Set 2 if available; else select set 0.
  - b. Set RF2 to 0--0100 0000(HEX) (SVA).
  - c. Start CPU at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Set MAC data output bit 18, in processor PTM register.
  - f. Set RF2 to 0--0100 0000(HEX) SVA with wrong parity.
  - g. Clear bit 18 in processor PTM register.
  - h. Set RFA to C300--0100 0000(HEX); write data.
  - i. Start processor at micrand 27, purge Cache, write word to SVA, wait for CPU halted.
  - j. Read CPU status, format and report word 1; error if not 080--0 (HEX); CARPE(2).
- C.7 CPU Status Word 2.
- a. Report word 2; error if not 0--08000000(HEX) (SVAPE).

80000285 A

LMT3/P - 147

# Conditions:

- C.8 Test byte three, use Set three or one, report CPU Status Word one.
- Select Cache Set 3 if available; else select set 1.
  - Set RF2 to 0--01 0000 0000(HEX) (SVA).
  - Start CPU at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set MAC data output bit 18, in processor PTM register.
  - Set RF2 to 0--01 0000 0000(HEX) SVA with wrong parity.
  - Clear bit 18 in processor PTM register.
  - Set RFA to C301--01 0000 0000(HEX); write data.
  - Start processor at micrand 27, purge Cache, write word to SVA, wait for CPU halted.
  - Read CPU status, format and report word 1; error if not 10--0 (HEX); CARPE(1).
- C.9 CPU Status Word 2.
- Report word 2; error if not 0--01000000(HEX) (SVAPE).
- C.10 Test byte two, use Set three or one, report CPU Status Word one.
- Set RF2 to 0--0100 0000 0000(HEX) (SVA).
  - Start CPU at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set MAC data output bit 18, in processor PTM register.
  - Set RF2 to 0--0100 0000 0000(HEX); SVA with wrong parity.
  - Clear bit 18 in processor PTM register.
  - Set RFA to C301--0100 0000 0000(HEX); write data.
  - Start processor at micrand 27, purge Cache, write word to SVA, wait for CPU halted.
  - Read CPU status, format and report word 1; error if not 20000020--0(HEX); CARPE(0) and CHTGPE.
- C.11 CPU Status Word 2.
- Report word 2; error if not 0--02000000 (SVAPE).
- C.12 CPU Status Word 1 From Allocates.
- Report word 1 if saved; error if reported.
- C.13 CPU Status Word 2 from Allocates.
- Report word 2 if saved; error if reported.
  - Set RF2 to 0--0; clear wrong parity data.
  - Purge all cache.
  - Start processor at Micrand 5, clear residue parity errors.
  - Deselect FAKECM mode.
- C.14 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-83; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

# Section 11, Subsection 4

This subsection tests the Cache Hit Code (CHC) generated by the CACHE HIT ANALYZER, to ensure the correct set number is returned when a data parity error is detected.

## Hardware Tested:

- Cache Allocate Network.

## Method:

The CHC is encoded from the Cache set match signals. The code is sent to OPI and accepted when a read data (to DAI) parity error is detected.

To force this situation, data with wrong parity is written to Cache. Then with a specific set selected, the data is read out generating a DAI parity error. The CHC should be accepted and passed on to PFS83.

## Test Patterns:

Refer to Conditions.

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |

## Initialization:

- Set Wrong Parity Data Into RFA.
  - Set bit 18 in processor PTM register; MAC parity out bit.
  - Set RFA to 0--01(HEX); write data, byte seven should have wrong parity.
  - Clear bit 18 in processor PTM register.
- Select FAKECM mode.

## Conditions:

- C.0 Cache Set selected (0, 1, 2, or 3).
- Select next Cache set.
  - Report set selected; informational.

Conditions:

- C.1 Allocate Cache, report CPU Status Word 1.  
a. Set RF2 to 0--0 (SVA).  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and report word 1; error if not 0.
- C.2 CPU Status Word 2.  
a. Report word 2; error if not 0.
- C.3 Write error data to Cache, report CPU Status Word 1.  
a. Start processor at micrand E, write word to SVA, wait for CPU halted.  
b. Read CPU status, format and report word 1; error if not 00010080--0(HEX); CHWDPE (7) and CHCPPE.
- C.4 CPU Status Word 2.  
a. Report word 2; error if not 0--010000(HEX); ADC A-Stream bit 7.
- C.5 Read data.  
a. Enable PDM error; clear bit 47 in processor DEC.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand 5, clear residue parity errors, wait for CPU halted.  
d. Start processor at micrand D, read word from SVA, wait for CPU halted.  
e. Read RF8, report data; error if not DF--DF(HEX); read data blocked.
- C.6 CPU Status Word 1.  
a. Disable PDM errors.  
b. Read CPU status, format and report word 1; error if not 0000040--0(HEX); SELRD(1).
- C.7 CPU Status Word 2.  
a. Report word 2; error if not 00x0000100010000(HEX). x = 1, 2, 4, or 8 depending on set selected; CHC code and DAI(7), and A-Stream (7).
- C.8 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Return to C.0 if:

- End of first pass and no errors.
- End of second pass with four sets available and no errors.
- End of fourth pass and no errors.

Deselect FAKECM mode.

Start processor at micrand 5, clear residue parity errors, wait for processor halted.

60000285 A

LMT3/P - 150

Section 11, Subsection 5

This subsection is kept as a space. Condition 0 is retained and reporting zero in order to maintain proper macro operation such as end of subsection.

60000285 A

LMT3/P - 151

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK DUE TO A LATE DELETION

THIS PAGE HAS BEEN INTENTIONALLY LEFT BLANK DUE TO A LATE DELETION



## Section 11, Subsection 6

This subsection tests the parity signals generated by the tag out parity checker.

### Hardware Tested:

- CRAMPE Signal Path.
- PFS83, bits 04 through 07.

### Method:

The CRAMPE Signal is forced by allocating cache using a SVA containing wrong parity. The wrong parity SVA is sent to Local Memory and entered into the Cache Tag RAM using a FAKECM Allocate. This is followed by a FAKECM Write operation using the same SVA, now with good parity, to effect the hit and bring the bad parity tag out of the tag RAM. The parity error should be sensed and the signal sent to MAC.

On the second pass the CRAMPE Signal is disabled. The same sequence (pass 1) is repeated. However, instead of selecting set 0, all sets except 0 are selected. The CRAMPE signal should not be sensed.

Passes one and two test sets 0 and 1 in a like manner. Passes three and four test sets 2 and 3. If only two sets are available only two passes are run.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| E      | Write Word to SVA.     |
| F      | Purge All Cache.       |

### Initialization:

- Select all Cache Sets.
- Select FAKECM Mode.
- Set Pass Count to 1.
- Set RFA to E7--E7(HEX); write data.

### Conditions:

- C.0 Pass count (1 through 4).
- a. Report pass count; informational.

60000285 B

LMT3/P - 154

### Conditions:

- C.1 Report CPU Status Word 1 After Allocate.
- a. Start processor at micrand F, purge All Cache, wait for CPU halted.
  - b. Select Cache set 0 (pass 1) or 2 (pass 3).
  - c. Set MAC Parity Data Out bit 18 in processor PTM register.
  - d. Set RF2 to 0--08000(HEX); SVA with wrong parity.
  - e. Clear bit 18 in processor PTM register.
  - f. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - g. Read CPU Status, format and report Word 1; error if not 020000250--0(HEX); CARPE,CHTBP,AMUXPEs.
- C.2 Report CPU Status Word 2 after allocate.
- a. Report Word 2; error if not 880--02000000(HEX) for pass 1 and 2 or 440--02000000(HEX) for pass 3 and 4; CADRPE,CRAMPE,SVAPPE from ADC.
- C.3 Report CPU Status Word 1 after write.
- a. Deselect current Cache set and select all other available sets.
  - b. Set RF2 to 0--08000(HEX); SVA, good parity.
  - c. Start processor at micrand 5, clear CAR/CABR registers, wait for CPU halted.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU Status, format and report Word 1; error if not 0.
- C.4 Report CPU Status Word 2 after write.
- a. Report Word 2; error if not 080--0(HEX) for pass 1 or 020--0(HEX) for pass 3 (CRAMPE), 0 for passes 2 and 4.
- C.5 Report CPU Status Word 1 after allocate.
- a. Start processor and Micrand F, purge all Cache, wait for CPU halted.
  - b. Select Cache set 0 (pass 1) or 2 (pass 3).
  - c. Set MAC parity data out bit (18) in processor PTM register.
  - d. Set RF2 to 0--08000(HEX); SVA with wrong parity.
  - e. Clear bit 18 in processor PTM register.
  - f. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - g. Read CPU Status, format and report Word 1; error if not 020000250--0(HEX); CARPE,CHGAP,AMUXPEs.
- C.6 Report CPU Status Word 2 after allocate.
- a. Report Word 2; error if not 440--02000000(HEX) for pass 1 and 2 or 110--02000000(HEX) for pass 3 and 4; CADRPE,CRAMPE,SVAPPE from ADC.

60000285 A

LMT3/P - 155

Conditions:

- 2.7 Report CPU Status Word 1 after write.
  - a. Deselect current Cache set and select all other available sets.
  - b. Set RF2 to 0--08000(HEX); SVA, good parity.
  - c. Start processor at micrand 5, Clear CAR/CABR registers, wait for CPU halted.
  - d. Start processor at micrand E, Write Word to SVA, wait for CPU halted.
  - e. Read CPU Status, format and report Word 1; error if not 0.
- 2.8 Report CPU Status Word 2 after write.
  - a. Report Word 2; error if not 04--0(HEX) for pass 1 or 01--0(HEX) for pass 3 (CRAMPE), 0 for passes 2 and 4.
- 2.9 CPU Status Word 1 from purge.
  - a. Report Word 1, if saved; error if reported.
- 2.10 CPU Status Word 2 from purge.
  - a. Report Word 2, if saved; error if reported.
- 2.11 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if four passes complete, or if only two sets available and two passes complete; else return to C.0.

Deselect FAKECM Mode.

Purge All Cache.

SECTION 12 - PURGE CACHE FUNCTIONS

This section tests the Purge Cache Block function. Purge 512 bytes (64 words) starting at an even 64-word multiple as specified by the address SVA sent to LM.

Test the purge Cache by ASID function. Purge all Cache words currently allocated having the same ASID as the SVA sent to LM by the purge instruction.

Subsection 0 - Purge Cache block (512 bytes).

Subsection 1 - Purge Cache by ASID.

Subsection 2 - Purge Cache by ASID continued.

Method:

Refer to subsection descriptions.

Initialization:

- Set Purge Micrand (Micrand 28) into Control Store at address 105(HEX).

Section 12, Subsection 0

This subsection tests that the Purge Cache Block functions correctly in all available sets. A Cache Block Purge (BLKPUR) will purge 512 bytes, which is 64 words or 16 Cache blocks.

Hardware Tested:

- Purge Block Decode.
- LM Counter (FH gate).
- Inhibit bits 55 through 58 to Mux C.
- BLKPUR to Cache Sets.

Method:

The three major areas tested in this subsection are tested as follows:

1. Block Purge Control: A purge code of 0 or 3 will initiate a block purge. Two passes use a code of 0, two passes use a code of 3. These codes come from the Segment Map only as a result of passing on the purge code from a virtual instruction. Therefore, Instruction Fetch (IF) must be used to start this micrand.

# Method (continued):

2. Inhibit bits 55 through 58 in Adrs Mux C: Among the four SVAs to be used, one per pass, bits 55 through 58 will be set to various patterns; zeros, ones and alternating bits. In all cases, bits 55 through 58 should be zeroes in the Adrs Mux and the purge address defined by Cache address bits 51 through 54.
3. BLKPUR Signal to Cache: Tested by sending each block purge to a different set. For example, pass 0 to set 0, pass 1 to set 1, and so forth. If only two sets exist they will be used twice.

Conditions 1 and 2 use FAKECM mode to allocate and write word zero of every location in the current Cache set.

Condition three then executes the Block Purge by setting the purge instruction into the adjacent cache set. For example, if set 0 is currently selected then set 1 is used to hold the purge instruction. Instruction Fetch (IF) is used to read the purge instruction from cache, word 0, in FAKECM Mode. The code is sent to Local Memory where the purge is performed.

Conditions 4 and 5 now read word zero from all locations written during Conditions 1 and 2. The data read is checked and should either be as written, if not in the purged block, or zero if within the 16 Tags (512 bytes) purged.

## Test Patterns (hexadecimal):

| Pass | Set    | Cache Tag<br>Purge SVA | Addresses Purged |
|------|--------|------------------------|------------------|
| 1    | 0      | 0000 0000 0000         | 000 to 00F       |
| 2    | 1      | 0000 0000 05E0         | 020 to 02F       |
| 3    | 2 or 0 | 0000 0000 0940         | 040 to 04F       |
| 4    | 3 or 1 | 0000 0000 0CA0         | 060 to 06F       |

Write Pattern = C3xx yyyy yyyy (HEX).

xx = Set number.  
yy = Current SVA.

## Micrands Used:

| Number | Description                  |
|--------|------------------------------|
| C      | FAKECM Allocate Cache.       |
| D      | Read Word from SVA.          |
| E      | Write Word to SVA.           |
| F      | Purge Cache All.             |
| 10     | Set P Descriptor register.   |
| 13     | Enter P Register (start IF). |
| 28     | Virtual Purge Micrand.       |

# Initialization:

- Purge Cache All.
- Start processor at micrand F, Purge Cache, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set RF10 to 0--010000(HEX); PVA for IF operation.
- Set P Descriptor to 0--0.
- Set RF11 to 0--0; segment number.
- Set CM(0) to 0--0; P Descriptor Value.
- Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
- Read CPU status, format and save if error, error if not zero, if error, set failing condition number to -1(FE(HEX)).
- Set current Cache Set to 0.
- Select FAKECM mode.
- Set Pass Count to 1.

## Conditions:

- C.0 Pass count (1 through 4).
  - a. Select current Cache set.
  - b. Set current SVA to 0.
  - c. Report pass count; informational.
- C.1 Allocate SVA in selected Set.
  - a. Set RF2 to current scan SVA.
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current scan SVA; informational.
- C.2 Write to current scan SVA.
  - a. Set RFA to current write data.
  - b. Start processor at micrand E, write word at SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current write data; informational.

Continue to next condition if error, SSM, or if 100(HEX) SVAs written; else advance SVA (+20(HEX) and return to C.1.

Conditions:

- C.3 Execute Block Purge.
- Select current and adjacent cache sets.
  - Set RF2 to 0--010000(HEX) cache tag for IF.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU Status, format and save if error; error if not 0.
  - Set RFA to 05x0--0(HEX); virtual purge instructions x = purge code.
  - Start processor at micrand E, write RFA to cache, wait for CPU halted.
  - Read CPU Status, format and save if error; error if not 0.
  - Set RF2 to current purge SVA.
  - Start processor at micrand I3, set P register (execute purge), wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set current scan SVA to 0.
  - Report purge instruction; informational.

- C.4 Read SVA.
- Report current scan SVA; informational.

- C.5 Read data from Cache.
- Set RF2 to current scan SVA.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0 for purged locations, or value written for non-purged locations.

Continue to next condition if error or if 100(HEX) SVAs read; else advance SVA (+20(HEX) and return to C.4.

- C.6 CPU Status Word 1.
- Report word 1 if saved; error if reported.

- C.7 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.8 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if four passes complete; else advance Cache set number and return to C.0. If sets 2 and 3 are not available, repeat sets 0 and 1.

Deselect FAKECM mode.

Section 12, Subsection 1

This subsection tests the Purge Cache operation when specific ASIDs are to be purged from Cache.

Hardware Tested:

- Purge Segment Control.
- LM Counter (CT04 Gate).
- Inhibit bits 51 through 58 of Adrs MUX.
- Purge Segment Signal to Cache Sets.

Method:

The Purge Segment operation is initiated when a purge code of 1, 4, 5, 6 or 7 is received. All codes accomplish the same result. The Purge Segment Control logic is tested by all the purge codes issued in sequential order.

Adrs Mux bits 51 through 58 are forced to zero by the hardware during the Purge Segment. This is tested by issuing SVAs with bit patterns set in that bit area, then ensuring that the purge started at Cache address zero.

The Purge Segment Signals to the individual Cache sets are tested by issuing at least one purge to each available Cache set.

Condition 0 selects the Cache set to test and reports the Pass Count, four passes.

Conditions 1 and 2 allocate and write, using the Scan SVAs, formatted data to word 0 of all 256 Cache blocks in the selected set.

Condition 3 performs the purge function using the next sequential purge code (1, 4, 5, 6, 7). The purge instruction, with the purge code embedded, is set into the adjacent cache set. For example, if set 0 is currently selected then set 1 is used to hold the purge instruction. Sets 2 and 3 are similarly used. Instruction Fetch (IF) is used to read the current purge instruction from cache word 0, in FAKECM Mode. The code is sent to LM where the purge is performed.

Conditions 4 and 5 provide the data check. These conditions are repeated 256 times reading word 0 from each Cache block, using the Scan SVA, and testing the data to ensure all words associated with the purge ASID have been purged.

The condition 3, 4, 5 sequence is repeated three times within each pass by Conditions 6, 7, 8, and 9, 10, 11 respectively. Each successive purge uses the next purge code and the next Purge SVA. See Test Patterns for details.

FAKECM mode is selected except during the actual purge.

# Test Patterns (hexadecimal):

Cache is initialized in C.1 and C.2 as follows.

Passes 1 and 3 Allocate and Write Set 0 and Set 2; if available; if not, repeat set 0, using the Scan SVAs.

Passes 2 and 4 Allocate and Write Set 1 and Set 3; if available; if not repeat set 1, using the Scan SVAs.

| Pass 1 or 3       |                  |   | Pass 2 or 4       |                |  |
|-------------------|------------------|---|-------------------|----------------|--|
| Cache Tag Address | Scan SVA         |   | Cache Tag Address | Scan SVA       |  |
| 0                 | 0000 0000 00000  |   |                   | FFFF 0000 0000 |  |
|                   |                  |   |                   |                |  |
| v                 | vv               | v |                   |                |  |
| 3F                | 0000 0000 07E03F |   |                   | FFFF 0000 07E0 |  |
| 40                | FFFF 0000 080040 |   |                   | 5555 0000 0800 |  |
|                   |                  |   |                   |                |  |
| v                 | vv               | v |                   |                |  |
| 7F                | FFFF 0000 0FE07F |   |                   | 5555 0000 0FE0 |  |
| 80                | 5555 0000 100080 |   |                   | AAAA 0000 1000 |  |
|                   |                  |   |                   |                |  |
| v                 | vv               | v |                   |                |  |
| BF                | 5555 0000 17E0BF |   |                   | AAAA 0000 17E0 |  |
| C0                | AAAA 0000 1800C0 |   |                   | 0000 0000 1800 |  |
|                   |                  |   |                   |                |  |
| v                 | vv               | v |                   |                |  |
| FF                | AAAA 0000 1FE0FF |   |                   | 0000 0000 1FE0 |  |

Write data = C3xx yyyy yyyy yyyy(HEX).

xx = Set number written.

yy = Current SVA.

| Pass | Condition Number | Purge Code | Purge SVA      |
|------|------------------|------------|----------------|
| 1    | 3                | 1          | 0000 0000 1FE0 |
| 1    | 6                | 4          | FFFF 0000 1540 |
| 1    | 9                | 5          | 5555 0000 0AA0 |
| 2    | 3                | 6          | AAAA 0000 0000 |
| 2    | 6                | 7          | 0000 0000 1FE0 |
| 2    | 9                | 1          | FFFF 0000 1540 |
| 3    | 3                | 4          | 5555 0000 0AA0 |
| 3    | 6                | 5          | AAAA 0000 0000 |
| 3    | 9                | 6          | 0000 0000 1FE0 |
| 4    | 3                | 7          | FFFF 0000 1540 |
| 4    | 6                | 1          | 5555 0000 0AA0 |
| 4    | 9                | 4          | AAAA 0000 0000 |

## Micrands Used:

| Number | Description                |
|--------|----------------------------|
| C      | FAKECM Allocate Cache.     |
| E      | Write Word to SVA.         |
| 13     | Set P Register (Start IF). |
| 28     | Virtual Purge Micrand.     |

## Initialization:

- Select FAKECM mode.
- Set RF10 to 0--010000(HEX); PVA for IF operation.
- Purge Cache All.
  - Start processor at micrand F, purge Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error, set failing condition number to -1(FE(HEX)).
- Set current Cache set to 0.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 4).
- Set current scan SVA to 0.
  - Select current Cache set.
  - Report pass count; informational.
- C.1 Allocate scan SVA in Selected Set.
- Set RF2 to current scan SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, if error, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.2 Write Data to Current scan SVA.
- Set RFA to current write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current write data; informational.

Continue to next condition if error, SSM, or if 100(HEX) SVAs have been written; else advance Scan SVA (+20(HEX) and return to C.1.

Conditions:

- C.3 Purge by ASID.
- Select current and adjacent cache sets.
  - Set RF2 to 0--0100000(HEX); cache tag.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU Status, error if not 0.
  - Set RFA to 050x0--0(HEX); virtual purge inst, x = purge code.
  - Start processor at micrand E, write RFA to cache, wait for CPU halted.
  - Read CPU Status, error if not 0.
  - Set RF2 to current purge SVA.
  - Start processor at micrand 13, set P register; initiates purge, wait for CPU halted.

1. Report purge instruction; informational.

C.4 Current read SVA.

- Report current scan SVA; informational.

C.5 Cache data.

- Set RF2 to current scan SVA.
- Set RF8 to DF--DF(HEX); filler.
- Start processor at micrand D, read word from SVA, wait for CPU halted.
- Read CPU status, format and save if error; error if not 0.
- Read RF8, report data; error if:
  - Not 0 for purged locations
  - Not value written for locations not purged.

Continue to next condition if error, SSM, or if 100(HEX) SVAs read; else advance Scan SVA (+20(HEX)) and return to C.4.

C.6 Purge by ASID.

- Select current and adjacent cache sets.
- Set RF2 to 0--0100000(HEX); cache tag.
- Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
- Read CPU Status, format and save if error; error if not 0.
- Set RFA to 050x0--0(HEX); virtual purge inst, x = purge code.
- Start processor at micrand E, write RFA to cache, wait for CPU halted.
- Read CPU Status, format and save if error; error if not 0.
- Set RF2 to current purge SVA.
- Start processor at micrand 13, set P register; initiates purge, wait for CPU halted.
- Read CPU status, format and save if error; error if not 0.
- Set scan SVA to 0.
- Report purge instruction; informational.

C.7 Current read SVA.

- Report current scan SVA; informational.

80000285 A

LMT3/P - 184

Conditions:

- C.8 Cache data.
- Set RF2 to current scan SVA.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if:
    - Not 0 for purged locations.
    - Not value written for locations not purged.

Continue to next condition if error, SSM, or if 100(HEX) SVAs read; else advance Scan SVA (+20(HEX)) and return to C.7.

C.9 Purge by ASID.

- Select current and adjacent cache sets.
- Set RF2 to 0--010000(HEX); cache tag.
- Start processor at micrand C, FAKECM Allocate Cache, wait for CPU halted.
- Read CPU Status, format and save if error; error if not 0.
- Set RFA to 050x0--0(HEX); virtual purge inst, x = purge code.
- Start processor at micrand E, write RFA to cache, wait for CPU halted.
- Read CPU Status, format and save if error; error if not 0.
- Set RF2 to current purge SVA.
- Start processor at micrand 13, set P register; initiates purge, wait for CPU halted.
- Read CPU status, format and save if error; error if not 0.
- Set scan SVA to 0.
- Report purge instruction; informational.

C.10 Current read SVA.

- Report current scan SVA; informational.

C.11 Cache data.

- Set RF2 to current scan SVA.
- Set RF8 to DF--DF(HEX); filler.
- Start processor at micrand D, read word from SVA, wait for CPU halted.
- Read CPU status, format and save if error; error if not 0.
- Read RF8, report data; error if:
  - Not 0 for purged locations.
  - Not value written for locations not purged.

Continue to next condition if error, SSM, or if 100(HEX) SVAs read; else advance Scan SVA (+20(HEX)) and return to C.10.

C.12 CPU Status Word 1.

- Report word 1 if saved; error if reported.

C.13 CPU Status Word 2.

- Report word 2 if saved; error if reported.

80000285 A

LMT3/P - 185

Conditions;

C.14 First Failure Capture Information.

- a. Report PFS87 bits 19-23 in bits 59-63; information only.
- b. Report last PTL written in bits 34-47; information only.
- c. Report last PTA written in bits 13-31; information only.

Exit Subsection if end of fourth pass; else advance set number and return to C.0.

Section 12, Subsection 2

This subsection continues to test the Purge Cache by ASID operation and also continues testing the data written into the Cache Tag Register over the CAR0 lines.

Hardware Tested:

- Purge Segment Control.
- LM Counter (CT04 Gate).
- Purge Segment Signal to Cache Sets.
- Cache Tag Ram logic for Cache Sets.

Method:

The Purge Segment operation is initiated by performing a virtual purge segment (0501) instruction in instruction fetch.

Various ASID and BN combinations within a SVA are used to test the CAR address lines and the cache tag rams associated with each cache set within local memory. After purging all of cache, a cache set will be allocated with the various combinations to be tested (while in FAKECM mode). A store word SVA operation will be performed on each allocated location and then followed by a load word SVA operation to ensure that the data was indeed written to cache. When all of cache and its operands have been verified, a test is made to determine if the cache tag ram purged ASID correctly. A purge is performed on one ASID at a time with all of the allocated SVAs being read again and tested for errors. The data obtained from the load word SVA micrand should be nonzero if its associated ASID was not purged and zero if its ASID had been previously purged.

Condition 0 selects the cache set to test and reports the pass count (1 through 4) which indicates which set was run.

Conditions 1, 2, and 3 allocate, write, and read using the scan SVA micrands.

Condition 4 performs the purge instruction which uses the purge code of 1 (0501). Instruction fetch is used to read the current instruction from cache after which the code is sent to local memory and the purge is performed.

Conditions 5 and 6 provide the data check after the purge. A load word SVA is performed on each SVA that was originally allocated in conditions 1 through 3. The data is tested to ensure that only the proper locations were purged in condition 4.

FAKECM mode is selected except during the actual purge.

Test Patterns (hexadecimal):

For use in Conditions 1 through 3  
and Conditions 5 through 6

For use in  
Condition 4

| Test Count 1* | Scan SVA       | Test Count 2** | Purge SVA      |
|---------------|----------------|----------------|----------------|
| 1.            | 0001 0000 0020 | 1.             | 0001 0000 0000 |
| 2.            | 0001 7FFF E040 | 2.             | 0002 0000 0000 |
| 3.            | 0001 2AAA A080 | .              | .              |
| 4.            | 0001 5555 2080 | .              | .              |
| 5.            | 0002 0000 00A0 | F.             | 4000 0000 0000 |
| 6.            | 0002 7FFF E0C0 | 10.            | 8000 0000 0000 |
| .             | .              | .              | .              |
| 3F.           | 8000 2AAA A7E0 |                |                |
| 40.           | 8000 5555 2800 |                |                |

- \* - Test count 1 is each pass through C.1 through C.3 during the allocate, write, and read SVA before going to C.4; or Test count 1 is each pass through C.5 performing a read SVA and verifying the data after a purge in C.4.
- \*\* - Test count 2 is each pass through C.4 where a purge by ASID is performed which is then followed by various passes in C.5.

Write data in C.2 = C3xx yyyy yyyy(HEX).

xx = Set number written.

yy = Current SVA.

Micrands Used:

| Number | Description                |
|--------|----------------------------|
| C      | FAKECM allocate cache.     |
| D      | Load word from SVA.        |
| E      | Write word to SVA.         |
| F      | Purge all cache.           |
| 13     | Set P register (start IF). |
| 28     | Virtual purge micrand.     |

#### Initialization:

- Select FAKECM mode.
- Set RF10 to 0--010000(HEX); PVA for IF operation.
- Set current cache set to 0.
- Set pass count to 1.

#### Conditions:

- C.0 Pass count 1 through 4.
- a. Purge cache all.
    - Start processor at micrand F, purge cache, wait CPU halted.
    - Read CPU status, format and save if error; error if not zero.
    - If error, set failing condition number to -1(FE(HEX)).
  - b. Select the current cache set.
  - c. Get scan SVA for pass 1 and put in buffer for use in C.1.
  - d. Report pass count; informational.
- C.1 Allocate scan SVA in selected set.
- a. Write RF02 with the current SVA for pass being performed.
  - b. Start processor at micrand C, FAKECM allocate, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current scan SVA used; informational.
- C.2 Write word to cache at current scan SVA.
- a. Create the write data by adding a C3xx(HEX) (xx = cache set) to the front of the scan SVA, and put the data in RF0A.
  - b. Start processor at micrand E, write SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report the write data; informational.
- C.3 Load word from cache at current scan SVA.
- a. Set RF08 to zero.
  - b. Start processor at micrand D, load SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read the contents of RF08 and determine if error in data.
    - If the data in RF08 = 0, report an error and display a expected pattern of DF--DF(HEX) as expected (should be nonzero).
    - If the data in RF08 is nonzero, report the data as good.

Continue to next condition if error, SSM, or all 40(HEX) SVAs have been allocated, written, and read; else, advance scan SVA to next pattern and return to C.1

#### Conditions:

- C.4 Execute a purge by ASID operation.
- a. If first time into C.4, set SVA for pass 1 for both the scan SVA and purge SVA in the appropriate buffers.
  - b. Write RF02 with 0-010000(HEX) (cache tag for purge instruction).
  - c. Start processor at micrand C, FAKECM allocate, wait CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Write RF0A with 05010--0(HEX), purge instruction.
  - f. Start processor at micrand E, write SVA, wait CPU halted.
  - g. Read CPU status, format and save if error; error if not 0.
  - h. Write RF02 with ASID to be purged for pass being run.
  - i. Start processor at micrand 13, start IF, wait for CPU halted.
  - j. Read CPU status, format and save if error; error if not 0.
  - k. Report the ASID purged; informational.
- C.5 Read all scan SVAs, and determine if error.
- a. If first time into C.5, set scan SVA to pass 1 SVA.
  - b. Write RF02 with appropriate scan SVA for pass being run.
  - c. Write RF08 with DF--DF(HEX); filler.
  - d. Start processor at micrand D, load SVA; wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF08 and determine if data error occurred.
    - If the data read was zero for a purged ASID, the location was good; report the zeros read.
    - If the data read was nonzero for a purged ASID, the location was bad; report the bad data read.
    - If the data read was zero and the ASID was not purged, the location was bad; report DF--DF(HEX) expected, but is actually indicating nonzero was expected.
    - If the data read was nonzero and the ASID was not purged, the location was good; report the data read.

Continue to next condition if error, SSM, or all load SVAs have been performed for all ASIDs purged; else, advance scan SVA and go to C.5, or advance to the next ASID to purge and go to C.4 after reset of scan SVA to 0000 001 0000 0020.

- C.6 Report scan SVA used in C.5.
- a. Report the scan SVA last used in C.5; informational.
- C.7 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit subsection if end of fourth pass; else advance set number and go to C.0.  
Deselect FAKECM mode.  
Reset Control Store address 105(HEX) to original micrand.



## SECTION 13 - CACHE ALLOCATE AND LRU NETWORK

This section tests both methods of allocating Cache: Set Not Valid and by LRU value. It also tests the LRU RAMs in all sets.

Subsection 0 - Cache Allocate by Set Not Valid.

Subsection 1 - Cache Status RAM (LRU code only) Data Test.

Subsection 2 - Cache LRU Update Network.

Subsection 3 - Status RAM Parity Signals.

Method:

Refer to subsection descriptions.

Initialization:

- Set PTA to 0.
- Set PTL to 0.
- Set PSM to 7F(HEX).

### Section 13, Subsection 0

This subsection tests the Cache Allocate Network using the Valid bit is clear situation.

Hardware Tested:

- Cache Allocate.

There are four steps required to test all sets of Allocate Not Valid logic.

1. All sets are allocated at block zero using unique SVAs.
2. Then starting with the last set allocated, block zero is purged. This clears the Valid bit and leaves all other sets Valid bits on, and their LRU Codes set to a nonzero value.
3. Allocation is again initiated using a unique SVA. This is the test and the set with the clear Valid bit allocated, even though all other sets LRU Codes are greater.
4. To determine if the expected set was allocated, all other sets are deselected and a write/read to the last SVA is executed. If a hit occurs and the written data is returned the test was successful.

60000285 A

LMT3/P - 170

Steps 2 through 4 are repeated for each available set.

The purge used is the Virtual Block Purge instruction. The instruction is set into Block 0 of Cache set 0. In Condition 10 Block 0 of Set 0 is purged, therefore the virtual instruction is again written in C.10 to allow looping.

Test Patterns (hexadecimal):

| Set | First SVA      | Second SVA     |
|-----|----------------|----------------|
| 0   | 0000 0000 0000 | 0000 0000 8000 |
| 1   | 0 ---- 0 2000  | 0 ---- 0 A000  |
| 2   | 0 ---- 0 4000  | 0 ---- 0 C000  |
| 3   | 0 ---- 0 6000  | 0 ---- 0 E000  |

Write data = C3xx yyyy yyyy yyyy(HEX).

xx = Set written to.  
yy = Current SVA.

Micrands Used:

| Number | Description                        |
|--------|------------------------------------|
| C      | FAKECM Allocate Cache.             |
| D      | Read Word From SVA.                |
| E      | Write Word to SVA.                 |
| 10     | Set P Descriptor register.         |
| 13     | Set P Register; start Purge Block. |
| F      | Purge Cache All.                   |
| 28     | Purge Micrand, Initiated from IF.  |

Initialization:

- Select all available Cache Sets.
- Set P Descriptor register to zero.
- Set RF11 to 0--0; Segment number.
- Set CM(0) to 0--0; P Descriptor Value.
- Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero, if error, set error condition number to -1(FE(HEX)).
- Purge Cache All.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero, if error, set error condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Set Purge Micrand (28) into Control Store at address 105(HEX).
- Set RF10 to 0--0; P Register Value.

60000285 A

LMT3/P - 171

Conditions:

- C.0 Allocate All Available Sets, Report Last SVA.
- Set RF2 to 0--0; set 0 SVA.
  - Set RFA to 050--0(HEX); virtual purge instruction.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand E, write purge instruction to Cache Set 0, wait for CPU halted.
  - Read CPU Status, format and save if error; error if not 0.
  - If error report SVA and exit condition.
  - Set RF2 to 0--02000(HEX); set 1 SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - If error report SVA and exit condition.
  - If only two sets available report SVA and exit condition.
  - Set RF2 to 0--04000(HEX); set 2 SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - If error report SVA and exit condition.
  - Set RF2 to 0--06000(HEX); set 3 SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA.
- C.1 Purge and reallocate Set 3, report allocated SVA.
- If only two sets available, report 0 and exit condition.
  - Set RF2 to 0--06000(HEX); purge SVA.
  - Start processor at micrand 13, set P register; initiate purge, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 0--0E000(HEX); new SVA.
  - Start processor at micrand C, FAKECM allocate Cache; actual test, Set 3 should be allocated, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report allocated SVA; informational.
- C.2 Write data to Set 3.
- If only two sets available, report 0 and exit condition.
  - Select Set 3 only.
  - Set RFA to formatted write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.

Conditions:

- C.3 Read data from Set 3.
- If only two sets available, report 0 and exit condition.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not as reported in C.2.
- C.4 Purge and reallocate Set 2, report allocated SVA.
- If only two sets available, report 0 and exit condition.
  - Select all available sets.
  - Set RF2 to 0--04000(HEX); Purge SVA.
  - Start processor at micrand 13, set P register; initiate purge, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 0--0C000(HEX); new SVA.
  - Start processor at micrand C, FAKECM allocate Cache; actual test, Set 2 should be allocated, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report allocated SVA; informational.
- C.5 Write data to Set 2.
- If only two sets available, report 0 and exit condition.
  - Select Set 2 only.
  - Set RFA to formatted write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.6 Read data from Set 2.
- If only two sets available, report 0 and exit condition.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not as reported in C.5.
- C.7 Purge and reallocate Set 1, report allocated SVA.
- Select all available sets.
  - Set RF2 to 0--02000(HEX); Purge SVA.
  - Start processor at micrand 13, set P register; initiate purge, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 0--0A000(HEX); new SVA.
  - Start processor at micrand C, FAKECM allocate Cache actual test, Set 1 should be allocated, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report allocated SVA; informational.

Conditions:

- C.8 Write data to Set 1.  
a. Select Set 1 only.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.9 Read data from Set 1.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 Purge and reallocate Set 0, report Allocated SVA.  
a. Select all available sets.  
b. Set RF2 to 0--00000(HEX); purge SVA.  
c. Set RFA to 050--0(HEX); virtual purge instruction.  
d. Start processor at micrand E, write purge instruction to Cache set 0, wait for CPU halted.  
e. Read CPU Status, format and save if error; error if not 0.  
f. Start processor at micrand 13, set P register; initiate purge, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Set RF2 to 0--08000(HEX) (new SVA).  
i. Start processor at micrand C, FAKECM allocate Cache; actual test, Set 0 should be allocated, wait for CPU halted.  
j. Read CPU status, format and save if error; error if not 0.  
k. Report allocated SVA; informational.
- C.11 Write data to Set 0.  
a. Select Set 0 only.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data; informational.
- C.12 Read data from Set 0.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.11.
- C.13 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

Conditions:

- C.15 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

Reset Control Store address 105(HEX) to original micrand.

Section 13, Subsection 1

This subsection tests all locations of each Set's Cache Status RAM to ensure LRU Code integrity and correct RAM addressing.

Hardware Tested:

- Cache Status RAMS.
- Cache Allocate.

Method:

The objective of this subsection is to initiate a series of events resulting in the LRU Code incrementing from 0 through 3 for each Tag address. These events are:

1. Four allocates using unique SVAs, all to the same Tag address. This will complete the Allocate Not Valid testing in all sets. If two sets are available only two allocates are required.
2. Another allocate to the same Tag address is issued. This allocate is by Highest LRU Code and Set 0 should be allocated.
3. To determine if Set 0 was allocated, it is individually selected and a write/read issued to the allocated SVA. If a miss occurred and zeros are returned, a failure (See Note) has occurred in the Cache Allocate Network or on one of the available Cache Tag boards.  
  
Steps 2 and 3 are repeated two or four times depending on the number of sets available.
4. A last allocate is performed to leave the LRU Codes such that if an address error should occur the expected sets will not be allocated correctly.
5. All SVAs are advanced to the next Cache block address +20(HEX) and steps 1 through 4 repeated until all 256 locations are tested.

# NOTE

Most errors should manifest themselves as Multi-Allocate errors.

FAKECM mode is used.

## Test Patterns:

| Set-Up             | SVAs           |
|--------------------|----------------|
| First Allocate     | 0000 0000 0xx0 |
| Second Allocate    | 0000 0000 2xx0 |
| Third Allocate**   | 0000 0000 4xx0 |
| Fourth Allocate**  | 0000 0000 6xx0 |
| Fifth Allocate     | 0000 0000 8xx0 |
| Sixth Allocate     | 0000 0000 Axx0 |
| Seventh Allocate** | 0000 0000 Cxx0 |
| Eighth Allocate**  | 0000 0000 Exx0 |
| Ninth Allocate     | 0000 0001 0xx0 |

\*\*Not executed when only two sets are available.

xx represents the 8-bit Cache Tag Address. It is shown right-shifted one place for clarity.

Write data = C3xx yyyy yyyy yyyy(HEX).  
xx = Set written.  
yy = SVA used.

One pass example using four sets:

|                                 | LRU Codes |       |       |       |
|---------------------------------|-----------|-------|-------|-------|
|                                 | Set 0     | Set 1 | Set 2 | Set 3 |
| After Cache Purge               | 0         | 0     | 0     | 0     |
| Condition 1 issues              | 0         | 1     | 1     | 1     |
| four allocates                  | 1         | 0     | 2     | 2     |
|                                 | 2         | 1     | 0     | 3     |
|                                 | 3         | 2     | 1     | 0     |
| Condition 2<br>allocates set 0  | 0         | 3     | 2     | 1     |
| Condition 4<br>allocates set 1  | 1         | 0     | 3     | 2     |
| Condition 6<br>allocates set 2  | 2         | 1     | 0     | 3     |
| Condition 8<br>allocates set 3  | 3         | 2     | 1     | 0     |
| Condition 10<br>allocates set 0 | 0         | 3     | 2     | 1     |

## Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word From SVA.    |
| E      | Write Word To SVA.     |
| F      | Purge Cache All.       |

## Initialization:

- Select all available Cache Sets.
- Purge Cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error, set error condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 100(HEX)).  
a. Report pass count; informational.
- C.1 Issue Set-Up allocates, report Last SVA.  
a. Set RF2 to 0--0xx0(HEX); first allocate SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. If error, report SVA and exit condition.  
e. Set RF2 to 0--02xx0(HEX); second allocate SVA.  
f. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. If error, report SVA and exit condition.  
i. If only two sets available, report SVA and exit condition.  
j. Set RF2 to 0--04xx0(HEX); third allocate SVA.  
k. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
l. Read CPU status, format and save if error; error if not 0.  
m. If error, report SVA and exit condition.  
n. Set RF2 to 0--06xx0(HEX); fourth allocate SVA.  
o. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
p. Read CPU status, format and save if error; error if not 0.  
q. Report SVA; informational.

Conditions:

- C.2 Allocate and write to Set 0, report write data.
  - a. Set RF2 to 0--08xx0(HEX); fifth allocate SVA.
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Set RFA to formatted write data.
  - e. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - f. Read CPU status, format and save if error; error if not 0.
  - g. Report write data; informational.
- C.3 Read data from Set 0.
  - a. Select Set 0 only.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not as reported in C.2.
- C.4 Allocate and Write to Set 1, Report Write Data.
  - a. Set RF2 to 0--0Axx0(HEX); sixth allocate SVA.
  - b. Select all available Cache sets.
  - c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Set RFA to formatted write data.
  - f. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - g. Read CPU status, format and save if error; error if not 0.
  - h. Report write data; informational.
- C.5 Read data from Set 1.
  - a. Select set 1 only.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not as reported in C.4.
- C.6 Allocate and write to Set 2, report write data.
  - a. If only two sets available, report 0 and exit condition.
  - b. Set RF2 to 0--0Cxx0(HEX); seventh allocate SVA.
  - c. Select all available Cache sets.
  - d. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Set RFA to formatted write data.
  - g. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Report write data; informational.

60000285 A

LMT3/P - 178

Conditions:

- C.7 Read data from Set 2.
  - a. If only two sets available, report 0 and exit condition.
  - b. Select Set 2 only.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF6, report data; error if not as reported in C.6.
- C.8 Allocate and write to Set 3, report write data.
  - a. If only 2 sets available, report 0 and exit condition.
  - b. Set RF2 to 0--0Exx0(HEX); eighth allocate SVA.
  - c. Select all available Cache sets.
  - d. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Set RFA to formatted write data.
  - g. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Report write data; informational.
- C.9 Read data from Set 3.
  - a. If only two sets available, report 0 and exit condition.
  - b. Select Set 3 only.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF6, report data; error if not as reported in C.8.
- C.10 Last allocate.
  - a. Select all available cache sets.
  - b. Set RF2 to 0--01 0xx0(HEX); ninth allocate SVA.
  - c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report SVA; informational.
- C.11 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) pass complete; else advance SVA's (+20(HEX) and return to C.0. Deselect FAKECM mode.

60000285 A

LMT3/P - 179

## Section 13, Subsection 2

This subsection tests the Cache LRU Update Network.

Test all available Cache sets for proper LRU update. Check the Reference LRU to Local LRU comparison. Test for LRU Clear after match, Advance if less than Reference LRU, and Unchanged if greater than Reference LRU.

### Hardware Tested:

- Cache Tag LRU Logic.
- Allocate, Reference LRU.

### Method:

The LRU update logic is tested by initializing each available Set's LRU Code at location zero. This is done by allocating all sets, resulting in LRU Codes as follows:

| For 2 Sets: |     | For 4 Sets: |     |
|-------------|-----|-------------|-----|
| Set         | LRU | Set         | LRU |
| 0           | 1   | 0           | 3   |
| 1           | 0   | 1           | 2   |
|             |     | 2           | 1   |
|             |     | 3           | 0   |

Cache block zero is used exclusively in this subsection.

A series of Cache write operations is now issued to various sets, followed by a single allocate. A write/read is issued to the set expected to have been allocated. If no data is returned, an error has occurred.

Condition one represents the sequence of operations and contains the Control Word. There can be up to four write operations and there is always one allocate/write/read operation. Condition one, the Control Word, is illustrated as:

C.1 = FFFF FFFF FFFv wxyz(HEX)

The order and Cache set numbers where the write data is expected to go are given in vwxy. If any of these hexadecimal characters is F, that particular write is skipped. The z is the set expected to be allocated.

Conditions 2, 3, 4 and 5 perform the write operations, some of which may be skipped depending on the number of sets available.

Conditions 6, 7 and 8 perform the allocate/write/read sequence. An error will usually appear as a Multiple Set Hit error.

60000285 A

LMT3/P - 180

This subsection is designed to function with either two or four Cache sets. When running with two sets, passes 1 and 2 test all the possible LRU combinations. Therefore they are executed and the subsection is exited.

Note that the LRU codes left at the end of a particular pass are not necessarily the same as the preceding pass. In other words, pass one initializes the codes for pass two, etc. One pass cannot be looped on with similar results expected.

FAKECM mode is selected.

### Test Patterns (hexadecimal):

| Pass | Control Word        | Pass | Control Word         |
|------|---------------------|------|----------------------|
| 1    | FFFF FFFF FFFF F010 | E    | FFFF FFFF FFFF 1201  |
| 2    | FFFF FFFF FFFF F101 | F    | FFFF FFFF FFFF 2012  |
| 3    | FFFF FFFF FFFF F202 | 10   | FFFF FFFF FFFF 3013  |
| 4    | FFFF FFFF FFFF F020 | 11   | FFFF FFFF FFFF 0130  |
| 5    | FFFF FFFF FFFF F303 | 12   | FFFF FFFF FFFF 1301  |
| 6    | FFFF FFFF FFFF F030 | 13   | FFFF FFFF FFFF 2312  |
| 7    | FFFF FFFF FFFF F121 | 14   | FFFF FFFF FFFF 3123  |
| 8    | FFFF FFFF FFFF F212 | 15   | FFFF FFFF FFFF 1231  |
| 9    | FFFF FFFF FFFF F313 | 16   | FFFF FFFF FFFF 02310 |
| A    | FFFF FFFF FFFF F131 | 17   | FFFF FFFF FFFF 23102 |
| B    | FFFF FFFF FFFF F232 | 18   | FFFF FFFF FFFF 31023 |
| C    | FFFF FFFF FFFF F323 | 19   | FFFF FFFF FFFF 10231 |
| D    | FFFF FFFF FFFF 0120 |      |                      |

SVAs used for allocation, writing and reading.

| Set | SVA            |
|-----|----------------|
| 0   | 0000 0000 0000 |
| 1   | 0000 0000 2000 |
| 2   | 0000 0000 4000 |
| 3   | 0000 0000 6000 |

Write data = C3xx yyyy yyyy(HEX).

xx = Set written.  
yy = SVA written.

### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word From SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |

60000285 A

LMT3/P - 181

# Initialization:

- Select all available Cache Sets.
- Purge Cache all.
  - Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error, set failing condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Allocate all available Sets.
  - Set RF2 to 0--0; Set 0 SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error set failing condition number to -1(FE(HEX)).
  - Set RF2 to 0--02000(HEX); Set 1 SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error set failing condition number to -1(FE(HEX)).
  - If two sets available, go execute conditions.
  - Set RF2 to 0--04000(HEX); Set 2 SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error set failing condition number to -1(FE(HEX)).
  - Set RF2 to 0--06000(HEX); Set 3 SVA.
  - Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero, if error set failing condition number to -1(FE(HEX)).
- Set pass count to 1.

# Conditions:

- C.0 Pass count (1 through 19(HEX).
  - a. Report pass count; informational.
- C.1 Control Word.
  - a. Get next Control Word.
  - b. If it is a Cache Select Code, save code and get next Control Word.
  - c. Select Cache Sets to test.
  - d. Report current Control Word; informational.
- C.2 First write, report data.
  - a. Determine set number of first write from Control Word. If F (HEX), report 0s and exit condition.
  - b. Set RF2 to selected sets SVA.
  - c. Set RFA to formatted write data.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Report write data; informational.

60000285 A

LMT3/P - 182

# Conditions:

- C.3 Second write, report data.
  - a. Determine set number of second write from Control Word. If F (HEX), report 0s and exit condition.
  - b. Set RF2 to selected sets SVA.
  - c. Set RFA to formatted write data.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Report write data; informational.
- C.4 Third write, report data.
  - a. Determine set number of third write from Control Word. If F (HEX), report 0s and exit condition.
  - b. Set RF2 to selected sets SVA.
  - c. Set RFA to formatted write data.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Report write data; informational.
- C.5 Fourth write, report data.
  - a. Determine set number of fourth write from Control Word. If F (HEX), report 0s and exit condition.
  - b. Set RF2 to selected sets SVA.
  - c. Set RFA to formatted write data.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Report write data; informational.
- C.6 Allocate SVA, report SVA.
  - a. Determine set number of allocate from sequence indicator.
  - b. Set RF2 to selected sets SVA.
  - c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report SVA; informational.
- C.7 Write to Cache Set, report write data.
  - a. Set RFA to formatted write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report write data; informational.
- C.8 Read data from expected Set, report data.
  - a. Select set allocated in C.6 only.
  - b. Set RF6 to DF----DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data, error if data not as reported in C.7.

60000285 A

LMT3/P - 183

#### Conditions:

- C.9 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 19(HEX) passes complete, or if only two Cache Sets available and two passes complete; else advance Control Word index.

Deselect FAKECM mode.

#### Section 13, Subsection 3

This subsection tests the Status Code Parity Signal (STSPE) by forcing parity errors and checking for the correct CPU status.

#### Hardware Tested:

- Cache Status PE Network.

#### Method:

The LRU Code force parity bit is set in the processor PTM register. This forces the parity bit of the 3-bit status code to zero. Several allocates are now executed to increment and clear the LRU status code. After the operation is complete the CPU status is read and the expected parity status is checked.

FAKECM mode is used.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read using SVA.        |
| F      | Purge Cache All.       |

60000285 A

LMT3/P - 184

#### Initialization:

- Select all available Cache Sets.

#### Conditions:

- C.0 Purge Cache all, report CPU Status Word 1.
  - a. Start processor at micrand F, purge Cache all, wait for CPU halted.
  - b. Read CPU status, format and report word 1; error if not 0.
- C.1 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.2 Allocate 1, expect no errors, report CPU Status Word 1.
  - a. Set the LRU force parity bit (bit 43) in processor PTM register.
  - b. Select FAKECM mode.
  - c. Set RF2 to 0--0 (SVA).
  - d. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - e. Read CPU status, format and report word 1; error if not 0.
- C.3 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.4 Allocate 2, expect no errors, report Word 1.
  - a. Set RF2 to 0--02000(HEX) (SVA).
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and report word 1; error if not 0.
- C.5 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.6 Allocate 3, errors in Set 3, report Word 1.
  - a. Set RF2 to 0--04000(HEX) (SVA).
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and report word 1; error if not 0.
- C.7 Report CPU Status Word 2.
  - a. Report word 2; error if not 0 (2 sets), or 010--0(HEX); CRAMPE (3), 4 sets.
- C.8 Allocate 4, errors in Sets 0 and 3, report Word 1.
  - a. Set RF2 to 0--06000(HEX) (SVA).
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and report word 1; error if not 0.
- C.9 Report CPU Status Word 2.
  - a. Report word 2; error if not 0 (2 sets), or 0909--0(HEX); CRAMPE (0,3), 4 sets.

60000285 A

LMT3/P - 185



Conditions:

- C.10 Read 1, errors in Sets 0, 3, report Word 1.  
a. Start processor at micrand D, read using SVA, wait for CPU halted.  
b. Read CPU status, format and report word 1; error if not 0.
- C.11 Report CPU Status Word 2.  
a. Report word 2; error if not 040--0(HEX); CRAMPE(1), 2 sets, or 090--0(HEX); CRAMPE(0, 3), 4 sets.
- C.12 Read 2, expect errors in Sets 0, 2, and 3, report Word 1.  
a. Set RF2 to 0--04000.  
b. Start processor at micrand D, read using SVA, wait for CPU halted.  
c. Read CPU status, format and report word 1; error if not 0.
- C.13 Report CPU Status Word 2.  
a. Report word 2; error if not 0C0--0(HEX); CRAMPE(0, 1) 2 sets, or 0B0--0(HEX); CRAMPE(0, 2, 3) 4 sets.
- C.14 Read 3, errors in Sets 0, 1, and 2, report Word 1.  
a. Set RF2 to 0--02000(HEX) (SVA).  
b. Start processor at micrand D, read using SVA, wait for CPU halted.  
c. Read CPU status, format and report word 1; error if not 0.
- C.15 Report CPU Status Word 2.  
a. Report word 2; error if not 080--0(HEX); CRAMPE(0) 2 sets, or 0E0--0(HEX); CRAMPE(0, 1, 2) 4 sets.

Clear bit 43 in processor PTM register.

Deselect FAKECM mode.

Purge All Cache to clear PEs.

SECTION 14 - ADDRESS CONTROL A STREAM COUNTER

The A Stream Address Counter (AADCT) is a component of Address Control and is used to increment the byte address. It is used during BDP stream operations and during byte loads or stores when the byte string is contained in two 64-bit CM words.

The low order bits of AADCT were tested by the Address Control Test ACT3, using RMA type addressing. This method limited the highest address to the Model 855 minimum memory configuration. Now with the SVA mechanism tested the upper bits of the counter can be tested.

Subsection 0 - Address Control A Stream Counter.

Method:

Refer to subsection description.

Initialization:

- Set PTA to 0.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Select all available Cache Sets.

Section 14, Subsection 0

This subsection tests the upper bits of the A Stream Address Counter (AADCT).

Hardware Tested:

- AADCT bits 16 through 27.

Method:

The AADCT is caused to increment when a byte store operation is issued using a byte address and byte length such that two CM words are required; word boundary condition. This causes the submitted SVA to be sent to LM with the first part of the byte string, the SVA is then incremented in the AADCT and the result is sent to LM with the second part of the byte string.

FAKECM mode is used to allocate Cache at the current SVA. The SVA is then incremented and a second allocate issued. A Byte Store operation is then submitted storing eight bytes across two Cache words. The two Cache words are set to a known pattern prior to the byte store. The pattern preset into Cache is the current pass count repeated 8 times at 8-bit (one-byte) intervals.

The Byte Store pattern is formatted as F0F1F2F3F4F5F6F7, and the starting byte address is always 7. Therefore, after the operation the two Cache words should contain:

Word 1 = xx xx xx xx xx xx xx F0    Word 2 = F1 F2 F3 F4 F5 F6 F7 xx

xx = Current pass count.

Test Patterns (hexadecimal):

| Pass | First SVA      | Second SVA     |
|------|----------------|----------------|
| 1    | 0000 0000 FFFF | 0000 0001 0000 |
| 2    | 0000 0001 FFFF | 0000 0002 0000 |
| 3    | 0000 0002 FFFF | 0000 0003 0000 |
| 4    | 0000 0003 FFFF | 0000 0004 0000 |
|      |                |                |
| v    | v              | v              |
| 10   | 0000 000F FFFF | 0000 0010 0000 |
| 11   | 0000 001F FFFF | 0000 0020 0000 |
|      |                |                |
| v    | v              | v              |
| 1F   | 0000 00FF FFFF | 0000 0100 0000 |
| 20   | 0000 01FF FFFF | 0000 0200 0000 |
| 21   | 0000 02FF FFFF | 0000 0300 0000 |
|      |                |                |
| v    | v              | v              |
| 2E   | 0000 0FFF FFFF | 0000 1000 0000 |
| 2F   | 0000 1FFF FFFF | 0000 2000 0000 |
|      |                |                |
| v    | v              | v              |
| 34   | 0000 8FFF FFFF | 0000 7000 0000 |
| 35   | 0000 0010 FFFF | 0000 0011 0000 |
| 36   | 0000 0020 FFFF | 0000 0021 0000 |
| 37   | 0000 0030 FFFF | 0000 0031 0000 |
|      |                |                |
| v    | v              | v              |
| 43   | 0000 00F0 FFFF | 0000 00F1 0000 |
| 44   | 0000 10FF FFFF | 0000 1100 0000 |
| 45   | 0000 20FF FFFF | 0000 2100 0000 |
| 46   | 0000 30FF FFFF | 0000 3100 0000 |
|      |                |                |
| v    | v              | v              |
| 4A   | 0000 70FF FFFF | 0000 7100 0000 |
| 4B   | 0000 0011 FFFF | 0000 0012 0000 |
| 4C   | 0000 0012 FFFF | 0000 0013 0000 |
|      |                |                |
| v    | v              | v              |
| 58   | 0000 001E FFFF | 0000 001F 0000 |
| 59   | 0000 11FF FFFF | 0000 1200 0000 |
| 5A   | 0000 12FF FFFF | 0000 1300 0000 |
|      |                |                |
| v    | v              | v              |
| 66   | 0000 1EFF FFFF | 0000 1FFF 0000 |

60000285 A

LMT3/P - 188

Write data = F0F1F2F3F4F5F6F7.

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word From SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |
| 11     | Write Byte to SVA.     |

Initialization:

- Purge Cache all.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero, if error, set error condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Set RF12 to 0--07(HEX); byte length.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 67(HEX).
  - a. Report pass count; informational.
- C.1 Allocate and report first SVA.
  - a. Set RF2 to current first SVA.
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current SVA; informational.
- C.2 Preset Cache at first word, report write data.
  - a. Set RFA to pcpcpc--pc. pc = current pass count; write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report write data; informational.
- C.3 Allocate and report second SVA.
  - a. Set RF2 to current second SVA.
  - b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report current SVA; informational.

60000285 A

LMT3/P - 189

# Conditions:

- C.4 Preset Cache at second word, report write data.
- Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report write data; informational.
- C.5 Actual result, first word.
- Set RFA to F0F1F2F3F4F5F6F7(HEX); byte write data.
  - Set RF2 to current first SVA.
  - Start processor at micrand 11, write byte to SVA; test micrand, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA; fetch word from Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data, error if not pc--pcF0(HEX); pc = current pass count.
- C.6 Actual result, second word.
- Set RF2 to current SVA+1.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not F1F2F3F4F5F6F7pc(HEX).
- C.7 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if error or if 67(HEX) passes complete; else advance SVA and return to C.0.

Deselect FAKECM mode.

## SECTION 15 - PAGE TABLE ADDRESS/LENGTH REGISTERS

Test data integrity of the Page Table Address (PTA) and the Page Table Length (PTL) registers using the Index Operation.

Subsection 0 - PTA/PTL Test, Part 1.

Subsection 1 - PTA/PTL Test, Part 2.

### Method:

Various test values are set into the PTA/PTL registers using a micrand sequence. The Index Operation is then used to form an RMA from the PTA/PTL and SVA. The SVA is held to zero during this test. The created RMA fetches a CM word which represents a Page Table Entry or Descriptor (PTD). All of CM is set to a unique pattern causing all Page Table references to be missed, and the Continue bit is clear terminating any page table search operation.

The Index information returned to the Register File contains the last RMA searched, the number of PTDs searched, and the Page Search Without Find (PSWF) status.

| 0     | 1 | 28  | 32         | 34  | 39 | 63  |
|-------|---|-----|------------|-----|----|-----|
| ----- |   |     |            |     |    |     |
| /     |   | / / |            | / / |    | / / |
| ----- |   |     |            |     |    |     |
| RMA   |   |     | PSWF COUNT |     |    |     |

The test PTA is masked to reflect the configured CM for the system being tested.

The Page Size Mask (PSM) is set to 7F(HEX), eliminating any shifting during Hash Code formation.

### Section 15, Subsection 0

This subsection tests the PTA and PTL registers using zeros and ones patterns.

### Hardware Tested:

- SPID Register, bits 16 through 47.
- SPID Register, bits 48 through 54.
- Init. PTA/PTL (IIIPT) Control.
- Index Control.
- Index Control.
- PTA/PTL registers.
- Init. PTA/PTL (IIIPT) MUX.
- RMA Mux Entry D.

# Method:

Refer to section description.

## Test Patterns (hexadecimal):

| Condition | PTA** | PTL  | SVA | RMA**  |
|-----------|-------|------|-----|--------|
| 0/1       | 00000 | 0000 | 0   | 000000 |
| 2/3       | 7FFFF | 3FFF | 0   | 7E0000 |
| 4/5       | 7FF00 | 0000 | 0   | 7E0000 |
| 6/7       | 7FFFF | 0000 | 0   | 7FFE00 |

\*PTA and RMA are shown as if used with a maximum CM configuration. The actual value used is masked to reflect the CM configuration of the system being tested.

## Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |
| 12     | Purge Map All.   |

## Initialization:

- Purge Map all.
- Start processor at micrand 12, purge MAP all, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FF(HEX)).
- Set all configured CM to zero.
- Set PSM to 7F(HEX).
- Set RF2 to 0--0 (SVA).
- Set all available CM to 0ACBD3120--0(HEX).

## Conditions:

- C.0 Expected Index Result for PTA equal to 0, PTL equal to 0.
- a. Set PTA equal to 0, set PTL equal to 0.
  - b. Report 0000 0000 8100 0000(HEX); informational.
- C.1 Actual Index Result.
- a. Set RF8 to DF--DF(HEX) filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.0.

## Conditions:

- C.2 Expected Index Result for PTA equal to 7FFFF(HEX), PTL equal to 3FFF(HEX).
- a. Set PTA equal to 7FFFF(HEX), set PTL equal to 3FFF(HEX).
  - b. Report FC00 0000 8100 0000(HEX); informational.
- C.3 Actual Index Result.
- a. Set RF8 to DF--DF(HEX) filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.2.
- C.4 Expected Index Result for PTA equal to 7C000(HEX), PTL equal to 0.
- a. Set PTA equal to F00(HEX), set PTL equal to 0.
  - b. Report FC00 0000 8100 0000(HEX); informational.
- C.5 Actual Index Result.
- a. Set RF8 to DF--DF(HEX) filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected Index Result for PTA equal to 7FFFF(HEX), PTL equal to 0.
- a. Set PTA equal to FFF(HEX), set PTL equal to 0.
  - b. Report FFFF F000 8100 0000(HEX); informational.\*\*
- C.7 Actual Index Result.
- a. Set RF8 to DF--DF(HEX) filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.6.
- C.8 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

## Section 15, Subsection 1

This subsection tests the PTA and PTL registers using alternate bit patterns.

### Hardware Tested:

- SPID Register, bits 18 through 47.
- SPID Register, bits 48 through 54.
- PTA/PTL registers.
- Init. PTA/PTL (IIPT) mux.
- RMA Mux Entry D.

### Method:

Refer to section description.

### Test Patterns (hexadecimal):

| Condition | PTA** | PTL  | SVA | RMA**   |
|-----------|-------|------|-----|---------|
| 0/1       | 55555 | 2AAA | 0   | AAAAA00 |
| 2/3       | 55555 | 1555 | 0   | A800000 |
| 4/5       | 2AAAA | 1555 | 0   | 5555400 |
| 6/7       | 2AAAA | 2AAA | 0   | 5000000 |

\*\*PTA and RMA are shown as if used with a maximum CM configuration. The actual value used is masked to reflect the CM configuration of the system being tested.

### Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |
| 12     | Purge Map All.   |

### Initialization:

- Purge Map all.
  - Start processor at micrand 12, purge Map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set RF2 to 0--0 (SVA).

### Conditions:

- C.0 Expected Index Result for PTA equal to 55555(HEX), PTL equal to 2AAA(HEX).
- a. Set PTA equal to 55555(HEX), set PTL equal to 2AAA(HEX).
  - b. Report 5555 5000 0100 0001(HEX); informational.

60000285 A

LMT3/P - 194

### Conditions:

- C.1 Actual Index Result.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.0.
- C.2 Expected Index Result for PTA equal to 55555(HEX), PTL equal to 1555(HEX).
- a. Set PTA equal to 555(HEX), set PTL equal to 55(HEX).
  - b. Report 5400 0000 0100 0001(HEX); informational.
- C.3 Actual Index Result.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.2.
- C.4 Expected Index Result for PTA equal to 2AAAA(HEX), PTL equal to 1555(HEX).
- a. Set PTA equal to 2AAAA(HEX), set PTL equal to 1555(HEX).
  - b. Report AAAA A000 0100 0001(HEX); informational.
- C.5 Actual Index Result.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected Index Result for PTA equal to 2AAAA(HEX), PTL equal to 2AAA(HEX).
- a. Set PTA equal to 2AAAA(HEX), set PTL equal to 2AAA(HEX).
  - b. Report A800 0000 0100 0001(HEX); informational.
- C.7 Actual Index Result.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.6.
- C.8 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.

60000285 A

LMT3/P - 195

Conditions:

- C.10 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

SECTION 16 - PAGE TABLE ADDRESS TO RMA MUX

This section tests the PTA/PTL/Hash Code logic network which forms the Page Table RMA. It also tests that entry to the RMA mux.

Subsection 0 - Initial PTA to RMA mux, Part 1.

Subsection 1 - Initial PTA to RMA mux, Part 2.

Method:

The method described in Section 15 is again used here. That is, this section uses the Index Operation to search a Page Table consisting of a unique pattern (no hits).

The RMA formation logic is again tested in this section. Now the PTA is held to a constant zero and the PTL and SVA are varied to generate different RMA test addresses. The Hash Code is controlled by the ASID portion of the SVA. The remaining bits in the SVA are left zero.

Initialization:

- Set all CM to 0ACBD3120--0(HEX), if Section 15 did not.

Section 16, Subsection 0

Initial PTA to RMA Mux Test, Part 1.

This subsection tests the PTA/PTL/Hash Code logic network. This network forms the RMA used to start a Page Table Search.

Hardware Tested:

- SPID/Hash Code Network.
- SPID/Hash Code Network.
- RMA Mux Entry D.

Method:

Refer to section description.

# Test Patterns (hexadecimal):

| Condition | PTL  | SVA       | Hash Code | RMA     |
|-----------|------|-----------|-----------|---------|
| 0/1       | 0000 | FFFF 0--0 | FFFF      | 00001FE |
| 2/3       | 3FFF | FFFF 0--0 | FFFF      | 07FFFFE |
| 4/5       | 3FFF | 5555 0--0 | 5555      | 02AAAAA |
| 6/7       | 3FFF | AAAA 0--0 | AAAA      | 0155554 |

## Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |
| 12     | Purge Map All.   |

## Initialization:

- Set PTA to zero.
- Purge Map All.
- Start processor at micrand 12, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).

## Conditions:

- C.0 Expected Index Result.
- a. Set PTL to 0.
  - b. Set RF2 to FFFF 0000 0000(HEX) (SVA).
  - c. Report 0000 0FF0 0100 0001(HEX); informational.
- C.1 Actual Index Result.
- a. Set RF8 to DF--DF; filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.0.
- C.2 Expected Index Result.
- a. Set PTL to 3FFF(HEX).
  - b. Report 03FF FFF0 0100 0001(HEX); informational.
- C.3 Actual Index Result.
- a. Set RF8 to DF--DF; filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.2.

## Conditions:

- C.4 Expected Index Result.
- a. Set RF2 to 5555 0000 0000(HEX) (SVA).
  - b. Report 0155 5550 0100 0001(HEX); informational.
- C.5 Actual Index Result.
- a. Set RF8 to DF--DF; filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected Index Result.
- a. Set RF2 to AAAA 0000 0000(HEX) (SVA).
  - b. Report 02AA AAA0 0100 0001(HEX); informational.
- C.7 Actual Index Result.
- a. Set RF8 to DF--DF; filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.6.
- C.8 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

## Section 18, Subsection 1

Initial PTA to RMA Mux Test, Part 2.

This subsection tests the PTA/PTL/Hash Code logic network. This network forms the RMA used to start a Page Table Search.

## Hardware Tested:

- SPID/Hash Code Network.
- SPID/Hash Code Network.
- RMA Mux Entry D.

# Method:

Refer to section description.

## Test Patterns (hexadecimal):

| Condition | PTL  | SVA            | Hash Code | RMA    |
|-----------|------|----------------|-----------|--------|
| 0/1       | 2AAA | 5555 0000 0000 | 5555      | 0000AA |
| 2/3       | 1555 | 5555 0000 0000 | 5555      | 2AAAAA |
| 4/5       | 1555 | AAAA 0000 0000 | AAAA      | 000154 |
| 6/7       | 2AAA | AAAA 0000 0000 | AAAA      | 155554 |

## Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

## Initialization:

- None.

## Conditions:

- C.0 Expected Index Result.  
a. Set PTL to 2AAA(HEX).  
b. Set RF2 to 5555 0000 0000(HEX) (SVA).  
c. Report 0000 0550 0100 0001(HEX); informational.
- C.1 Actual Index Result.  
a. Set RF8 to DF--DF; filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.0.
- C.2 Expected Index Result.  
a. Set PTL to 55(HEX).  
b. Report 0155 5550 0100 0001(HEX); informational.
- C.3 Actual Index Result.  
a. Set RF8 to DF--DF; filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.2.

## Conditions:

- C.4 Expected Index Result.  
a. Set RF2 to AAAA 0000 0000(HEX) (SVA).  
b. Report 0000 0AA0 0100 0001(HEX); informational.
- C.5 Actual Index Result.  
a. Set RF8 to DF--DF; filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected Index Result.  
a. Set PTL to AA(HEX).  
b. Report 02AA AAA0 0100 0001(HEX); informational.
- C.7 Actual Index Result.  
a. Set RF8 to DF--DF; filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.6.
- C.8 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.



## SECTION 17 - HASH CODE LOGIC NETWORK

This section tests the shift and exclusive OR logic network that forms the Page Table Address Hash Code. It also completes testing of the Page Size Mask (PSM).

Subsection 0 - Test the Hash Shifter.

Subsection 1 - Test the PTA Exclusive OR (EXCL OR) logic.

### Method:

The mechanism used in this section to test for correct RMA generation is the Index operation to a unique pattern Page Table method described earlier. Specific methods are described in the individual subsections.

### Initialization:

- Set all CM to 0ACBD3120--0(HEX), if Sections 15 or 16 did not.
- Set PTA equal to 0.
- Set PTL equal to FF(HEX).

### Section 17, Subsection 0

This subsection tests the Hash Code mux shifter and part of the exclusive OR network. It also tests the remaining untested bits of the SPID register and the PSM input lines and catching register.

### Hardware Tested:

- Hash Code Network.
- Hash Code Network.

### Method:

Input to the exclusive OR network comes from the SPID and from the Mux Shifter Rank B. The shift count has been kept constant by keeping the PSM register a constant 7F(HEX). This subsection will vary the PSM register through all eight possible counts starting at 7F(HEX), ending at 00. This varies the shift count from 0 through 8 respectively.

Eight passes are required to complete the test, one per PSM value. Each pass tests the shift network using 0s, Fs, and both alternate bit patterns.

| Pass | Condition | PSM | SVA            | Hash |
|------|-----------|-----|----------------|------|
| 1    | 2/3       | 7F  | 0000 7E00 01F8 | 0000 |
| 1    | 4/5       | 7F  | 0000 01FF FE00 | FFFF |
| 1    | 6/7       | 7F  | 0000 0155 5400 | AAAA |
| 1    | 8/9       | 7F  | 0000 00AA AA00 | 5555 |
| 2    | 2/3       | 7E  | 0000 7C00 03F8 | 0000 |
| 2    | 4/5       | 7E  | 0000 03FF FC00 | FFFF |
| 2    | 6/7       | 7E  | 0000 02AA A800 | AAAA |
| 2    | 8/9       | 7E  | 0000 0155 5400 | 5555 |
| 3    | 2/3       | 7C  | 0000 7800 07F8 | 0000 |
| 3    | 4/5       | 7C  | 0000 07FF F800 | FFFF |
| 3    | 6/7       | 7C  | 0000 0555 5000 | AAAA |
| 3    | 8/9       | 7C  | 0000 02AA A800 | 5555 |
| 4    | 2/3       | 78  | 0000 7000 0FF8 | 0000 |
| 4    | 4/5       | 78  | 0000 0FFF F000 | FFFF |
| 4    | 6/7       | 78  | 0000 0AAA A000 | AAAA |
| 4    | 8/9       | 78  | 0000 0555 5000 | 5555 |
| 5    | 2/3       | 70  | 0000 6000 1FF8 | 0000 |
| 5    | 4/5       | 70  | 0000 1FFF E000 | FFFF |
| 5    | 6/7       | 70  | 0000 1555 4000 | AAAA |
| 5    | 8/9       | 70  | 0000 0AAA A000 | 5555 |
| 6    | 2/3       | 60  | 0000 4000 3FF8 | 0000 |
| 6    | 4/5       | 60  | 0000 3FFF C000 | FFFF |
| 6    | 6/7       | 60  | 0000 2AAA 8000 | AAAA |
| 6    | 8/9       | 60  | 0000 1555 4000 | 5555 |
| 7    | 2/3       | 40  | 0000 0000 7FF8 | 0000 |
| 7    | 4/5       | 40  | 0000 7FFF 8000 | FFFF |
| 7    | 6/7       | 40  | 0000 5555 0000 | AAAA |
| 7    | 8/9       | 40  | 0000 2AAA 8000 | 5555 |
| 8    | 2/3       | 00  | 0000 0000 FFF8 | 0000 |
| 8    | 4/5       | 00  | 0000 7FFF 0000 | FFFF |
| 8    | 6/7       | 00  | 0000 2AAA 0000 | AAAA |
| 8    | 8/9       | 00  | 0000 5555 0000 | 5555 |

### Cond. Expected Results

|     |      |      |      |      |
|-----|------|------|------|------|
| 2/3 | 0000 | 0000 | 8100 | 0000 |
| 4/5 | 03FF | FFF0 | 8100 | 0000 |
| 6/7 | 02AA | AAA0 | 8100 | 0000 |
| 8/9 | 0155 | 5550 | 8100 | 0000 |

Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

Initialization:

- Set PSM value to 7F(HEX).
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 8).
  - a. Report pass count; informational.
- C.1 Current PSM value.
  - a. Set PSM register to current PSM value.
  - b. Report current PSM value; informational.
- C.2 Expected result, shift 0s.
  - a. Report expected result; informational.
- C.3 Actual result, shift 0s.
  - a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF6, report data; error if not as reported in C.2.
- C.4 Expected result, shift 1s.
  - a. Report expected result; informational.
- C.5 Actual result, shift 1s.
  - a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF6, report data; error if not as reported in C.4.
- C.6 Expected result, shift alternate bits.
  - a. Report expected result; informational.
- C.7 Actual result, shift alternate bits.
  - a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF6, report data; error if not as reported in C.6.
- C.8 Expected result, shift alternate bits.
  - a. Report expected result; informational.
- C.9 Actual result, shift alternate bits.
  - a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF6, report data; error if not as reported in C.8.

Conditions:

- C.10 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if eight passes complete; else adjust PSM value and return to C.0.

Section 17, Subsection 1

This subsection tests both inputs to the PTA HASH (EXCL OR) network using patterns not previously used.

Hardware Tested:

- PTA HASH (EXCL OR) Network.
- PTA HASH (EXCL OR) Network.

Method:

Prior to this subsection one of the two inputs to the PTA HASH (EXCL OR) has been held to zero (actually ones due to inversion). This subsection varies both areas of the SVA sent to the exclusive OR network.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

Initialization:

- Set PSM to 7F(HEX).

60000285 A

LMT3/P - 206

Conditions:

- C.0 Expected result, ones/ones, 1s test.
  - a. Report 0000 0000 8100 0001(HEX); informational.
- C.1 Actual result.
  - a. Set RF2 to FFFF 01FF FE00(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF8, report data; error if not as reported in C.0.
- C.2 Expected result, 5's and 1's test.
  - a. Report 02AA AAA0 8100 0001(HEX); informational.
- C.3 Actual result.
  - a. Set RF2 to 5555 01FF FE00(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF8, report data; error if not as reported in C.2.
- C.4 Expected result, AA's and 1's test.
  - a. Report 0155 5550 8100 0001(HEX); informational.
- C.5 Actual result.
  - a. Set RF2 to AAAA 01FF FE00(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected result, 1's and 5's test.
  - a. Report 0155 5550 8100 0001(HEX); informational.
- C.7 Actual result.
  - a. Set RF2 to FFFF 0155 5400(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - e. Read RF8, report data; error if not as reported in C.6.
- C.8 Expected result, 1's and AA's test.
  - a. Report 02AA AAA0 8100 0001(HEX); informational.

60000285 A

LMT3/P - 207

Conditions:

- C.9 Actual result.  
a. Set RF2 to FFFF 00AA AA00(HEX) (SVA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand 14, Index, wait for CPU halted.  
d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
e. Read RF6, report data; error if not as reported in C.8.
- C.10 CPU Status Word 1.  
a. Report word 1 of saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 of saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

SECTION 18 - THE RMA INCREMENTER TEST

This section tests the RMA Incrementer using operands that will ensure all logic is functioning correctly. This section does not test decrement.

Subsection 0 - Search up to 32 entries.

Subsection 1 - Complete the Incrementer Test.

Method:

The method employed in this section is the same as that used in previous sections. That is, it checks the result of an Index Operation to determine if the RMA generated is correct.

Refer to subsection description for individual subsection detail.

Initialization:

- Set all CM to 0ACBD3120--0(HEX), if sections 15, 16, or 17 did not.
- Set PTA to 0.
- Set PTL to 0.
- Set PSM to 7F(HEX).

Section 18, Subsection 0

This subsection searches up to 32 Page Table entries. It tests the RMA Incrementer by counting sequentially from 0 through 31.

Hardware Tested:

- RMA Incrementer.
- INCC Count of 32.
- PSWF due to Count equal to 32.

Method:

The Index Operation is used to search a pseudo Page Table located at CM(0). Initially all control bits in this Page Table are zero, resulting in search termination due to a clear Continue bit. After each Index operation the Continue bit (bit 1) is set in the PTD just used. The next Index operation will search one entry further, etc. This procedure is repeated until the Continue bit is set in 35 PTD locations. The RMA and INCC counts should advance until 32 Continues are encountered. Beyond 32 the search is terminated due to the INCC count being 32.

The search operation is repeated 35 times as a matter of convenience. There are five tests per pass and seven passes (an even 35).

# Test Patterns:

Refer to Conditions.

# Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

# Initialization:

- Set RF2 to 0--0 (SVA).
- Set CM 0--FF(HEX) to 0ACBD3120--0(HEX).
- Set pass count to 1.

# Conditions:

- C.0 Pass count (1 through 7).  
a. Report pass count; informational.
- C.1 Expected result for search of 1, 6, and so on.  
a. Report expected result for:  
pass 1 = 0000 0000 8100 0000(HEX) (search 1); informational only.
- 2 = 0000 0028 8600 0000(HEX) (search 6); informational only.  
3 = 0000 0050 8800 0000(HEX) (search 8(HEX)); informational only.  
4 = 0000 0078 9000 0000(HEX) (search 10(HEX)); informational only.  
5 = 0000 00A0 9500 0000(HEX) (search 15(HEX)); informational only.  
6 = 0000 00C8 9A00 0000(HEX) (search 1A(HEX)); informational only.  
7 = 0000 00F0 9F00 0000(HEX) (search 1F(HEX)); informational only.
- C.2 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Skip next step during first pass.  
c. Set Continue bit in PTD just used.  
d. Start processor at micrand 14, Index, wait for CPU halted.  
e. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
f. Read RF8, report data; error if not as reported in C.1.

60000285 A

LMT3/P - 210

# Conditions:

- C.3 Expected result for Search of 2, 7, and so on.  
a. Report expected result for:  
pass 1 = 0000 0008 8200 0000(HEX) (search 2); informational only.
- 2 = 0000 0030 8700 0000(HEX) (search 7); informational only.  
3 = 0000 0058 8C00 0000(HEX) (search C(HEX)); informational only.  
4 = 0000 0080 9100 0000(HEX) (search 11(HEX)); informational only.  
5 = 0000 00A8 9800 0000(HEX) (search 18(HEX)); informational only.  
6 = 0000 00D0 9B00 0000(HEX) (search 1B(HEX)); informational only.  
7 = 0000 00F8 A000 0000(HEX) (search 20(HEX)); informational only.
- C.4 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Set Continue bit in PTD just used.  
c. Start processor at micrand 14, Index, wait for CPU halted.  
d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
e. Read RF8, report data; error if not as reported in C.3.
- C.5 Expected result for search of 3, 8, and so on.  
a. Report expected result for:  
pass 1 = 0000 0010 8300 0000(HEX) (search 3); informational only.
- 2 = 0000 0038 8800 0000(HEX) (search 8); informational only.  
3 = 0000 0060 8D00 0000(HEX) (search D(HEX)); informational only.  
4 = 0000 0088 9200 0000(HEX) (search 12(HEX)); informational only.  
5 = 0000 00B0 9700 0000(HEX) (search 17(HEX)); informational only.  
6 = 0000 00D8 9C00 0000(HEX) (search 1C(HEX)); informational only.  
7 = 0000 00F8 A000 0000(HEX) (search 20(HEX)); informational only.
- C.6 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Set Continue bit in PTD just used.  
c. Start processor at micrand 14, Index, wait for CPU halted.  
d. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
e. Read RF8, report data; error if not as reported in C.5.
- C.7 Expected result for search of 4, 9, and so on.  
a. Report expected result for:  
pass 1 = 0000 0018 8400 0000(HEX) (search 4); informational only.
- 2 = 0000 0040 8900 0000(HEX) (search 9); informational only.  
3 = 0000 0068 8E00 0000(HEX) (search E(HEX)); informational only.  
4 = 0000 0090 9300 0000(HEX) (search 13(HEX)); informational only.  
5 = 0000 00B8 9800 0000(HEX) (search 18(HEX)); informational only.  
6 = 0000 00E0 9D00 0000(HEX) (search 1D(HEX)); informational only.  
7 = 0000 00F8 A000 0000(HEX) (search 20(HEX)); informational only.

60000285 A

LMT3/P - 211

# Conditions:

- C.8 Actual result.
- Set RF8 to DF--DF(HEX); filler.
  - Set Continue bit in PTD just used.
  - Start processor at micrand 14, Index, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF8, report data; error if not as reported in C.7.
- C.9 Expected result for Search of 5, A, and so on.
- Report expected result for:
    - pass 1 = 0000 0020 8500 0000(HEX) (search 5); informational only.
    - 2 = 0000 0048 8A00 0000(HEX) (search A(HEX)); informational only.
    - 3 = 0000 0070 8F00 0000(HEX) (search F(HEX)); informational only.
    - 4 = 0000 0098 9400 0000(HEX) (search 14(HEX)); informational only.
    - 5 = 0000 00C0 9900 0000(HEX) (search 19(HEX)); informational only.
    - 6 = 0000 00E8 9E00 0000(HEX) (search 1E(HEX)); informational only.
    - 7 = 0000 00F8 A000 0000(HEX) (search 20(HEX)); informational only.

- C.10 Actual result.
- Set RF8 to DF--DF(HEX); filler.
  - Set Continue bit in PTD just used.
  - Start processor at micrand 14, Index, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF8, report data; error if not as reported in C.9.
- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if seven passes complete; else return to C.0.

## Section 18, Subsection 1

Complete testing of the RMA Incrementer by ensuring all carries and enables are functional.

## Hardware Tested:

- RMA Incrementer.

## Method:

The Index Operation is used as explained in Subsection 0. Test RMAs have been generated using the Incrementer equations and should test all gates (except decrement).

The continue bit is set in four sequential locations starting at the test RMA. This allows the incrementer to search up to RMA+3 or across the boundary being tested. The rest of CM is always zero. During pass 34 (last pass) the boundary being tested is the upper limit of the Page Table Length. Therefore wraparound occurs and the resultant RMA is CM address 2.

## Test Patterns (hexadecimal):

| Pass | RMA (word address)<br>(IF PTL = FF (HEX)) | SVA<br>SVA     |
|------|-------------------------------------------|----------------|
| 1    | 00001E                                    | 000F 0000 0000 |
| 2    | 00002E                                    | 0017 0000 0000 |
| 3    | 00003E                                    | 001F 0000 0000 |
| 4    | 00005E                                    | 002F 0000 0000 |
| 5    | 00006E                                    | 0037 0000 0000 |
| 6    | 00007E                                    | 003F 0000 0000 |
| 7    | 00008E                                    | 005F 0000 0000 |
| 8    | 0000DE                                    | 006F 0000 0000 |
| 9    | 0000EE                                    | 0077 0000 0000 |
| A    | 0000FE                                    | 007F 0000 0000 |
| B    | 000038                                    | 001B 0000 0000 |
| C    | 000078                                    | 003B 0000 0000 |
| D    | 0001FE                                    | 00FF 0000 0000 |
| E    | 0002FE                                    | 017F 0000 0000 |
| F    | 0003FE                                    | 01FF 0000 0000 |
| 10   | 0005FE                                    | 02FF 0000 0000 |
| 11   | 0006FE                                    | 037F 0000 0000 |
| 12   | 0007FE                                    | 03FF 0000 0000 |
| 13   | 0008FE                                    | 05FF 0000 0000 |
| 14   | 000DFE                                    | 06FF 0000 0000 |
| 15   | 000EFE                                    | 077F 0000 0000 |
| 16   | 000FFE                                    | 07FF 0000 0000 |
| 17   | 00037E                                    | 018F 0000 0000 |
| 18   | 00017E                                    | 008F 0000 0000 |
| 19   | 00018E                                    | 00DF 0000 0000 |
| 1A   | 00038E                                    | 010F 0000 0000 |
| 1B   | 0003DE                                    | 01EF 0000 0000 |
| 1C   | 0003EE                                    | 01F7 0000 0000 |
| 1D   | 0007F8                                    | 03FB 0000 0000 |
| 1E   | 000FF8                                    | 07FB 0000 0000 |
| 1F   | 001FFE                                    | 0FFF 0000 0000 |
| 20   | 002FFE                                    | 17FF 0000 0000 |

| Pass | RMA (word address) | SVA            |
|------|--------------------|----------------|
| 21   | 003FFE             | 1FFF 0000 0000 |
| 22   | 005FFE             | 2FFF 0000 0000 |
| 23   | 008FFE             | 37FF 0000 0000 |
| 24   | 007FFE             | 3FFF 0000 0000 |
| 25   | 008FFE             | 5FFF 0000 0000 |
| 26   | 00DFFE             | 6FFF 0000 0000 |
| 27   | 00EFFE             | 77FF 0000 0000 |
| 28   | 00FFFE             | 7FFF 0000 0000 |
| 29   | 0037FE             | 18FF 0000 0000 |
| 2A   | 0077FE             | 38FF 0000 0000 |
| 2B   | 0017FE             | 08FF 0000 0000 |
| 2C   | 0018FE             | 0DFF 0000 0000 |
| 2D   | 001DFE             | 0EFF 0000 0000 |
| 2E   | 001EFE             | 0F7F 0000 0000 |
| 2F   | 001FF8             | 0FF8 0000 0000 |
| 30   | 00F7FE             | 78FF 0000 0000 |
| 31   | 01F7FE             | FFFF 0000 0000 |
| 32   | 017FFE             | F8FF 0000 0000 |
| 33   | 018FFE             | DFFF 0000 0000 |
| 34   | 01FFFE             | FFFF 0000 0000 |

The expected result is formatted as follows:

00xx xxxx 8500 0000(HEX)

Where xxxx is equal to the current SVAs ASID plus 2.

The expected result for pass 1 would be:

0000 0011 8500 0000(HEX).

Due to the hardware wrap of the address when exceeding the PTL size, and also due to the possible variances of central memory sizes, the expected result for iteration 34 will be either of the following:

0000 0000 8300 0000    on 128Mbyte increments  
or  
0000 0010 8500 0000    on other memory sizes

Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

80000285 B

LMT3/P - 214

#### Initialization:

- Set PTL to 3FFF(HEX) or maximum allowable value.
- Set current SVA to 000F 0000 0000(HEX).
- Set current RMA to 00001E(HEX).
- Set CM 0 through 3FFF(HEX) to 0ACBC3120--0(HEX).
- Set pass count to 1.

#### Conditions:

- C.0 Pass count (1 through 34(HEX).
  - a. Report pass count; informational.
- C.1 Current SVA.
  - a. Set RF2 to current SVA.
  - b. Report current SVA; informational.
- C.2 Expected result.
  - a. Set Continue bit in 4 words of CM starting at the current RMA.
  - b. Format and report expected result; informational.
- C.3 Actual result.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read RF8, report data; error if not as reported in C.2.
- C.4 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.5 CPU Status Word 2.
  - a. Report word 2; error if not 0000040--0(HEX) (PSWF).
- C.6 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 34(HEX) passes complete; else advance SVA and return to C.0.

80000285 A

LMT3/P - 215

SECTION 19 - RMA MUX - INCREMENTER LOGIC MERGE

This section tests the PTA/PTL/RMA Incrementer logic merge to the RMA mux. This input to the RMA mux is used during Page Table search activity and during a Block Fill operation.

Subsection 0 - RMA mux, Adder Merge - Part 1.

Subsection 1 - RMA mux, Adder Merge - Part 2.

Method:

The method used in this section is the same basic method described in the previous section. That is, the result of the Index operation is checked to determine if the RMA and number of entries searched is correct.

Refer to subsection descriptions for additional detail.

Initialization:

- Set all CM to 0ACBD3120--0(HEX), if Sections 15, 16, 17, or 18 did not.

Section 19, Subsection 0

This subsection tests the PTA/PTL/Adder merge entry to the RMA mux. Part one tests RMA Mux bits 0 through 19 through the logic merge.

Hardware Tested:

- Adder to RMA Mux Network.

Method:

The continue bit is set in the first entry of the Page Table. This will allow the adder to increment the computed Page Table Address and then stop due to no Continue bit. The PTA, PTL, and ASID parts of the SVA are set up with 0's, F's, 5's, and A's patterns. The PTA register is masked, limiting the CM test addresses to the configuration under test.

Test Patterns:

Refer to Conditions.

60000285 A

LMT3/P - 216

NOTE

The values shown in the conditions are set up to be used with a system having a CM configuration of 16 megabytes. The actual value used is masked to fit the system being tested.

Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

Initialization:

- Set PSM to 7F(HEX).
- Set RF2 to 0--0 (SVA).

Conditions:

- C.0 Expected result, PTA equal to Fs, PTL equal to Fs, ASID equal to 0s.  
a. Set PTA to 7FFFF(HEX) or maximum allowable value.  
b. Set PTL to 3FFF(HEX) or maximum allowable value.  
c. Set bit 1; continue in appropriate CM location; i.e. 1E0000(HEX).  
d. Format and report expected result, 00F0 0008 8200 0000(HEX);==  
Informational.
- C.1 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.0.
- C.2 Expected result, PTA equal to Fs, PTL equal to 0s, ASID equal to 0s.  
a. Set PTL to 00.  
b. Clear Continue bit in CM locations written in C.0.  
c. Set bit 1; Continue in next appropriate CM location; i.e. 1FFE00(HEX).  
d. Format and report expected result, 00FF F008 8200 0000(HEX);==  
Informational.
- C.3 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.2.

==The expected results shown are for a 16 megabyte configuration.

60000285 B

LMT3/P - 217



Conditions:

- C.4 Expected result, PTA equal to Fs, PTL equal to 5s, ASID equal to 0s.  
a. Set PTL to 55555(HEX), or maximum allowable value.  
b. Clear Continue bit in CM location written in C.2  
c. Set bit 1; Continue in next appropriate CM location;  
i.e. 1F5400(HEX).  
d. Format and report expected result, 00FA A008 8200 0000(HEX);==  
informational.
- C.5 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0  
or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.4.
- C.6 Expected result, PTA equal to Fs, PTL equal to As, ASID equal to 0s.  
a. Set PTL to 2AAAA(HEX) or maximum allowable value.  
b. Clear Continue bit in CM location written in C.4.  
c. Set bit 1; Continue in next appropriate CM location;  
i.e. CM 1EAA00(HEX).  
d. Format and report expected result, 00F5 5008 8200 000(HEX);==  
informational.
- C.7 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0  
or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.6.
- C.8 Expected result, PTA equal to 5s, PTL equal to 0s, ASID equal to 0s.  
a. Set PTA to 55555(HEX) or maximum allowable value.  
b. Set PTL to 00.  
c. Clear Continue bit in CM location written in C.6.  
d. Set bit 1; Continue in next appropriate CM location,  
i.e. CM AAA00(HEX).  
e. Format and report expected result, 0055 5008 8200 0000(HEX);==  
informational.
- C.9 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0  
or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.8.

==The expected results shown are for a 16 megabyte configuration.

60000285 B

LMT3/P - 218

Conditions:

- C.10 Expected result, PTA equal to As, PTL equal to 0s, ASID equal to 0s.  
a. Set PTA to 2AAAA(HEX) or maximum allowable value.  
b. Clear Continue bit in CM location written in C.8.  
c. Set bit 1; Continue in next appropriate CM location;  
i.e. CM 155400(HEX).  
d. Format and report expected result, 00AA A008 8200 0000(HEX);==  
informational.
- C.11 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0  
or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.10.
- C.12 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.13 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.14 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Clear CM 155400(HEX).

Section 19, Subsection 1

This subsection completes testing of the Adder entry to the RMA mux. Part two tests RMA Mux bits 12 through 28 through the logic merge.

Hardware Tested:

- Adder to RMA Mux Network.

Method:

The same method as described in Part 1 is used. The PTA register now remains a constant zero allowing PTL to select the adder output in forming the next RMA.

Various values are used in the PTL and ASID part of the SVA to get the desired adder output. Test patterns include 0s, Fs, 5s, and As.

Test Patterns:

Refer to Conditions.

==The expected results shown are for a 16 megabyte configuration.

60000285 B

LMT3/P - 219

Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

Initialization:

- Set PTA to 0--0.

Conditions:

- C.0 Expected result, PTL equal to 0s, ASID equal to Fs.  
a. Set PTL to 00.  
b. Set RF2 to FFFF 0000 0000(HEX)(SVA).  
c. Set bit 1; continue in CM 1FE(HEX).  
d. Format and report expected result, 0000 0FF8 8200 0000(HEX); informational.
- C.1 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.0.
- C.2 Expected result, PTL equal to Fs, ASID equal to Fs.  
a. Set PTL to FF(HEX).  
b. Clear Continue bit in CM 1FE(HEX).  
c. Set Continue bit in CM 1FFFE(HEX).  
d. Format and report expected result, 000F FFF8 8200 0000(HEX); informational.
- C.3 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.2.
- C.4 Expected result, PTL equal to As, ASID equal to FFAA(HEX).  
a. Set PTL to AA(HEX).  
b. Set RF2 to FFAA 0000 0000(HEX)(SVA).  
c. Clear Continue bit in CM 1FFFE(HEX).  
d. Set Continue bit in CM 15554(HEX).  
e. Format and report expected result, 000A AAA8 8200 0000(HEX); informational.
- C.5 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.4.

30000285 A

LMT3/P - 220

Conditions:

- C.6 Expected result, PTL equal to 5s, ASID equal to FF55(HEX).  
a. Set PTL to 55(HEX).  
b. Set RF2 to FF55 0000 0000(HEX)(SVA).  
c. Clear Continue bit in CM 15554(HEX).  
d. Set Continue bit in CM AAAA(HEX).  
e. Format and report expected result, 0005 5558 8200 0000(HEX); informational.
- C.7 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.6.
- C.8 Expected result, PTL equal to Fs, ASID equal to 5s.  
a. Set PTL to FF(HEX).  
b. Set RF2 to 5555 0000 0000(HEX)(SVA).  
c. Format and report expected result, 0005 5558 8200 0000(HEX); informational.
- C.9 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.8.
- C.10 Expected result, PTL equal to Fs, ASID equal to As.  
a. Set PTL to FF(HEX).  
b. Set RF2 to AAAA 0000 0000(HEX)(SVA).  
c. Clear Continue bit in CM AAAA(HEX).  
d. Set Continue bit in CM 15554(HEX).  
e. Format and report expected result, 000A AAA8 8200 0000(HEX); informational.
- C.11 Actual result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF8, report data; error if not as reported in C.10.
- C.12 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.13 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

80000285 A

LMT3/P - 221

Conditions:

- C.14 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Clear Continue bit in CM 15554(HEX).

SECTION 20 - PTA/PTL PARITY CIRCUITS

This section tests the PTA/PTL register parity checkers and parity signals.

Subsection 0 - PTA/PTL Parity Checkers.

Subsection 1 - PTA/PTL and LM Counter Parity Signals.

Method:

Subsection 0 tests the parity checkers to ensure no false parity errors are generated.

Subsection 1 tests the parity signals from LM to MAC to ensure parity errors can be detected should they occur.

Section 20, Subsection 0

This subsection tests the PTA and PTL parity checkers to ensure false errors do not occur.

Hardware Tested:

- PTA/PTL Parity Checkers.

Method:

To test the PTA Parity Checker a counting pattern is used. The PTA Parity Checker is organized into three segments.

|       |     |       |    |
|-------|-----|-------|----|
| 0     | 7 8 | 15 16 | 18 |
| ----- |     |       |    |
|       |     |       |    |
| ----- |     |       |    |

Because the largest segment is 8 bits the counting pattern must also extend 8 bits or from 0 through FF(HEX). The segment containing bits 0-7 is written with the same data pattern 0-FF(HEX) as the segment containing bits 8-15. The segment containing bits 16-18 contains the lower 3 bits that are in the other two segments.

The PTL register is a total of 14 bits and has 1 parity bit. Therefore, it must be entered with incrementing patterns from 0 through FF(HEX) in the largest portion and then with an incrementing pattern from 0 through 3F(HEX) in the final or shorter portion.

|       |     |    |
|-------|-----|----|
| 0     | 7 8 | 13 |
| ----- |     |    |
|       |     |    |
| ----- |     |    |

Conditions 4 and 5 test to insure that a write of the PTA or PTL register is not enabled unless micrand bit 29 is set. A load PTA/PTL micrand is written to CS location 90(HEX) that has micrand bit 29 clear. In condition 4, an attempt to write the PTA and PTL register is made while PTM bit 18 is set. The resulting data with parity errors should not get to the PTA or PTL register. The status checked in condition 4 only reports an error if PFS 82, bit 48 or 49 is set and the status in condition 5 is for information only.

The registers are entered and the CPU status sampled, no errors are expected.

Test Patterns (hexadecimal):

| Pass | PTL  | PTA    |
|------|------|--------|
| 1    | 0000 | 000000 |
| 2    | 0041 | 000809 |
| 3    | 0082 | 001012 |
| 4    | 00C3 | 00181B |
| 5    | 0104 | 002024 |
| 6    | 0145 | 00282D |
| ↓    | ↓    | ↓      |
| ↓    | ↓    | ↓      |
| 100  | 3FFF | 07FFFF |

Micrands Used:

None (except those needed to write the PTA/PTL registers).

Initialization:

- Write a load PTA/PTL micrand at CS location 90(HEX) but with FU micrand bit 29 clear.
- Set current PTA and PTL to 0.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 100(HEX)).
  - a. Report pass count; informational.
- C.1 Current PTL/PTA pattern.
  - a. Report current PTL and PTA pattern, bits 0 through 31 hold PTA, bits 32 through 63 hold PTL, both right justified; informational.
- C.2 Test current pattern, report CPU Status Word 1.
  - a. Set PTA with current pattern.
  - b. Set PTL with current pattern.
  - c. Read CPU status, format and report word 1; error if not 0.

Conditions:

- C.3 Report CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.4 Test current pattern; report CPU Status Word 1.
  - a. Set PTM bit 18.
  - b. Write register file with current pattern.
  - c. Clear PTM bit 18.
  - b. Transfer register file to PTA/PTL.
  - e. Read CPU status, format, and report word 1; error if PFS 82 bits 48 or 49 set.
- C.5 Report CPU Status Word 2.
  - a. Report word 2; information only.
- C.6 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 100(HEX) passes complete; else advance PTL and PTA and return to C.0.

Section 20, Subsection 1

This subsection tests the parity error signals from the PTA parity checker (PTAPE) and from the PTL Parity Checker (PTLPE). It also tests the Parity Error signal for the INCC count returned during Index or Test.

Hardware Tested:

- PTA/PTL Parity Circuits.
- PFS82 bits 48 and 49.
- INCC Parity Signal.

Method:

The MAC Data Output bit in the processor PTM register is set allowing data with wrong parity to be set into the PTA and PTL registers. The parity status is then read and the expected status bits checked.

The INCC Parity Generator signal is tested by selecting the Complement INCC (CPINCC) bit in the processor PTM register. Index is used to cause the parity error, first with an odd INCC count, and then with an even INCC count.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

#### Initialization:

- Set CM addresses 0 through 10(HEX) to 0ACBD3120--0(HEX).

#### Conditions:

- C.0 Test PTA bits 0 through 7 Parity Logic, report CPU Status Word 1.  
a. Set MAC Data Output bit 18 in processor PTM register.  
b. Set RF2A to 0000 0000 0010 0100(HEX).  
c. Clear bit 18 in processor PTM register.  
d. Set PTA register to contents of RF2A.  
e. Read CPU status, format and report word 1; error if not 00040--04000(HEX); PTAPE, CWDPE(5).
- C.1 Report CPU Status Word 2.  
a. Report word 2; error if not 0--040400(HEX); A-Stream(7) PE.
- C.2 Test PTA bits 8 through 15 Parity Logic, CPU Status Word 1.  
a. Set bit 18 in processor PTM register.  
b. Set RF2A to 0000 0000 0008 0200(HEX).  
c. Clear bit 18 in processor PTM register.  
d. Set PTA register to contents of RF2A.  
e. Read CPU status, format and report word 1; error if not 00040--04000(HEX); PTAPE, CWDPE(5).
- C.3 Report CPU Status Word 2.  
a. Report word 2; error if not 0-040400; A-Stream (7) PE.
- C.4 Test PTA bits 16 through 18 Parity Logic, CPU Status Word 1.  
a. Set bit 18 in processor PTM register.  
b. Set RF2A to 0000 0000 0004 2000(HEX).  
c. Clear bit 18 in processor PTM register.  
d. Set PTA register to contents of RF2A.  
e. Read CPU status, format and report word 1; error if not 00060--04000(HEX); PTAPE, CWDPE(5,6).
- C.5 Report CPU Status Word 2.  
a. Report word 2; error if not 0--060600(HEX); A-Stream(6,7) PE.
- C.6 Test PTL bits 0-7, report CPU Status Word 1.  
a. Clear PTA and residue PEs.  
b. Set bit 18 in processor PTM register.  
c. Set RF2A to 0000 0000 0002 0100(HEX).  
d. Clear bit 18 in processor PTM register.  
e. Set PTL register to contents of RF2A.  
f. Read CPU status, format and report word 1; error if not 00040--08000(HEX); PTLPE, CWDPE(5).
- C.7 Report CPU Status Word 2.  
a. Report word 2; error if not 0--040400(HEX); A-Stream(7) PE.

#### Conditions:

- C.8 Test PTL bits 8-13 Parity Logic; report CPU Status Word 1.  
a. Set bit 18 in processor PTM register.  
b. Set RF2A to 0000 0000 0001 0400(HEX).  
c. Clear bit 18 in processor PTM register.  
d. Set PTL register to contents of RF2A.  
e. Read CPU status, format and report word 1; error if not 00060--08000(HEX); PTLPE, CWOPE(5,6).
- C.9 Report CPU Status Word 2.  
a. Report word 2; error if not 0--060600(HEX); A-Stream(6,7) PE.
- C.10 INCC Parity Generator, odd address, report Word 1.  
a. Set PTA to 0.  
b. Set PTL to 0.  
c. Clear residue PEs.  
d. Set RF2 to 0 (SVA).  
e. Set bit 22 in the processor PTM Register (CPINCC).  
f. Start processor at micrand 14, Index, wait for CPU halted.  
g. Read CPU status, format and report Word 1; error if not 0000020--0(HEX); SELRD(2).
- C.11 Report CPU Status Word 2.  
a. Report Word 2; error if not 000004080--0(HEX); PSWF and DA1(4).
- C.12 INCC Parity Generator, even address, report Word 1.  
a. Set Continue bit 1 in CM 0.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and report Word 1; error if not 0000020--0(HEX); SELRD(2).
- C.13 Report CPU Status Word 2.  
a. Report Word 2; error if not 000004080--0(HEX); PSWF and DA1(4).
- C.14 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.
- Clear bit 22 in processor PTM register.

## SECTION 21 - PAGE TABLE HIT COMPARATOR

This section tests the SPID/Page Table Entry compare network. This 38-bit Comparator indicates a Page Table Hit when parts of the SVA (SPID) match the Page Table Descriptor Tag sent from CM as a result of a Map Miss and Page Table search.

Subsection 0 - Page Table Hit Comparator - Match Valid.

Subsection 1 - Page Table Hit Comparator - Match Not Valid.

### Method:

A Page Table Descriptor (PTD) is set up in CM at an address determined by the patterns used. This PTD has the Valid, Continue, Used, and Modify bits set and the Tag portion set up to match the incoming SVA. The Page Frame Address (PFA) is set to zero, and the word following the PTD is cleared.

The Index operation is used to initiate a Page Table Search. If the match is valid, the Index Status word returned will have the RMA of the hit, the number of entries searched set to 1, and the PSWF flag clear. If the search results in no compare, the Index Status word returned will have the RMA of the Miss + 1, the number of entries searched set to 2, and the PSWF flag set.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Set CM 0 through 2000(HEX) to 0ACBD3120--0(HEX).

### Section 21, Subsection 0

This subsection tests the Page Table Hit Comparator for Match Valid. It tests all 38 bits of the SPID/PTD Comparator when expecting hit indications.

### Hardware Tested:

- CM to Comparator Network.
- Comparator Network.
- Page Table Hit Control.
- Map Control.

### Method:

This subsection tests the Comparator to ensure hits that should occur do occur. Patterns of 0s, Fs, 5s, and As are used with all 38 bits.

60000285 A

LMT3/P - 228

The Comparator hardware logic is split into two pieces, half the bits are compared on Map Set 0, the remaining half on Map Set 1. By setting up the patterns such that conditions 1/2 and 3/4 compare the same pattern on both boards and conditions 5/6 and 7/8 compare opposite patterns on both boards the failing indications can be used to determine which set is at fault.

Two passes are required to use all test patterns.

### Test Patterns (hexadecimal):

| Pass | Conditions | Pattern Type | Pattern to Board 0 | Board 1 | SVAs**         |
|------|------------|--------------|--------------------|---------|----------------|
| 1    | 1/2        | 0/0          | 0--0               | 0--0    | 0000 0000 0000 |
| 1    | 3/4        | F/F          | F--F               | F--F    | FFFF 7FFF FE00 |
| 1    | 5/6        | 0/F          | 0--0               | F--F    | 0000 7FFF 8E00 |
| 1    | 7/8        | F/0          | F--F               | 0--0    | FFFF 0000 7000 |
| 2    | 1/2        | 5/5          | 5--5               | 5--5    | 5555 2AAA AA00 |
| 2    | 3/4        | A/A          | A--A               | A--A    | AAAA 5555 5400 |
| 2    | 5/6        | 5/A          | 5--5               | A--A    | 5555 5555 2400 |
| 2    | 7/8        | A/5          | A--A               | 5--5    | AAAA 2AAA DA00 |

\*\*SVA bit 32 is actually discarded before the SVA data is sent to the Comparator.

### Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

### Initialization:

- Set pass count to 1.

### Conditions:

- C.0 Pass count (1 or 2).
- a. Report pass count; informational.

60000285 A

LMT3/P - 229

Conditions:

- C.1 Expected result, 0/0 pass 1, 5/5 pass 2.  
a. Set RF2 to 0000 0000 0000 0000(HEX) (pass 1 SVA).  
0000 5555 2AAA AA00(HEX) (pass 2 SVA).  
b. Set CM 1000(HEX) to F000 0000 0000 0000(HEX) (pass 1 PTD).  
CM 1000(HEX) to F555 5555 5540 0000(HEX) (pass 2 PTD).  
c. Format and report expected result; informational  
pass 1 = 0000 8000 0100 0000(HEX).  
pass 2 = 0000 8000 0100 0000(HEX).
- C.2 Actual result, 0/0 pass 1, 5/5 pass 2.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.1.
- C.3 Expected result, F/F pass 1, A/A pass 2.  
a. Set RF2 to 0000 FFFF 7FFF FE00(HEX) (pass 1 SVA).  
0000 AAAA 5555 5400(HEX) (pass 2 SVA).  
b. Set CM 1000(HEX) to FFFF FFFF FFC0 0000(HEX) (pass 1 PTD).  
CM 1000(HEX) to FAAA AAAA AA80 0000(HEX) (pass 2 PTD).  
c. Format and report expected result; informational  
pass 1 = 0000 8000 0100 0000(HEX).  
pass 2 = 0000 8000 0100 0000(HEX).
- C.4 Actual result, F/F pass 1, A/A pass 2.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.3.
- C.5 Expected result, 0/F pass 1, 5/A pass 2.  
a. Set RF2 to 0000 0000 7FFF 8E00(HEX) (pass 1 SVA).  
0000 5555 5555 2400(HEX) (pass 2 SVA).  
b. Set CM 118E(HEX) to F000 0FFF F1C0 0000(HEX) (pass 1 PTD).  
CM 118E(HEX) to F555 5AAA A2A0 0000(HEX) (pass 2 PTD).  
c. Format and report expected result; informational  
pass 1 = 0000 8C70 0100 0000(HEX).  
pass 2 = 0000 8C70 0100 0000(HEX).
- C.6 Actual result, 0/F pass 1, 5/A pass 2.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.5.

60000285 A

LMT3/P - 230

Conditions:

- C.7 Expected result, F/0 pass 1, A/5 pass 2.  
a. Set RF2 to 0000 FFFF 0000 7000(HEX) (pass 1 SVA).  
0000 AAAA 2AAA DA00(HEX) (pass 2 SVA).  
b. Set CM 118E(HEX) to FFFF F000 0E00 0000(HEX) (pass 1 PTD).  
CM 118E(HEX) to FAAA A555 5B40 0000(HEX) (pass 2 PTD).  
c. Format and report expected result; informational.  
pass 1 = 0000 8C70 0100 0000(HEX).  
pass 2 = 0000 8C70 0100 0000(HEX).
- C.8 Actual result, F/0 pass 1, A/5 pass 2.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 14, Index, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.7.
- C.9 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if two passes complete; else return to C.0.

Section 21, Subsection 1

This subsection tests the Page Table Hit Comparator, Match Not Valid. It tests the 38-bit SPID/PTD Comparator network when expecting Miss indications.

Hardware Tested:

- Comparator Network.

Method:

This subsection tests each bit separately to ensure a false hit does not occur. A sliding bit pattern is used with both zeros and ones.

If a failure occurs during passes 1 through 16 or 33 through 35 the errors most likely occurred on Map Set 0 board. Similarly, if pass 17 through 32 or 36, 37, or 38 should fail, suspect the board for Map Set 1.

60000285 A

LMT3/P - 231

Test Patterns (hexadecimal):

| Pass | Address<br>of PTD (C.1) | Positive<br>PTD (C.2) | Positive SVA<br>for C.4 |
|------|-------------------------|-----------------------|-------------------------|
| 1    | 1002                    | F000 0000 0040 0000   | 0000 0000 0200          |
| 2    | 1004                    | F000 0000 0080 0000   | 0000 0000 0400          |
| 3    | 1008                    | F000 0000 0100 0000   | 0000 0000 0800          |
| 4    | 1010                    | F000 0000 0200 0000   | 0000 0000 1000          |
| 5    | 1020                    | F000 0000 0400 0000   | 0000 0000 2000          |
| 6    | 1040                    | F000 0000 0800 0000   | 0000 0000 4000          |
| 7    | 1080                    | F000 0000 1000 0000   | 0000 0000 8000          |
| 8    | 1100                    | F000 0000 2000 0000   | 0000 0001 0000          |
| 9    | 1000                    | F000 0000 4000 0000   | 0000 0002 0000          |
| A    | 1000                    | F000 0000 8000 0000   | 0000 0004 0000          |
|      |                         |                       |                         |
| v    | v                       | v                     | v                       |
| 16   | 1000                    | F000 0800 0000 0000   | 0000 4000 0000          |
| 17   | 1002                    | F000 1000 0000 0000   | 0001 0000 0000          |
| 18   | 1004                    | F000 2000 0000 0000   | 0002 0000 0000          |
| 19   | 1008                    | F000 4000 0000 0000   | 0004 0000 0000          |
| 20   | 1010                    | F000 8000 0000 0000   | 0008 0000 0000          |
| 1B   | 1020                    | F001 0000 0000 0000   | 0010 0000 0000          |
| 1C   | 1040                    | F002 0000 0000 0000   | 0020 0000 0000          |
| 1D   | 1080                    | F004 0000 0000 0000   | 0040 0000 0000          |
| 1E   | 1100                    | F008 0000 0000 0000   | 0080 0000 0000          |
| 1F   | 1000                    | F010 0000 0000 0000   | 0100 0000 0000          |
| 20   | 1000                    | F020 0000 0000 0000   | 0200 0000 0000          |
|      |                         |                       |                         |
| v    | v                       | v                     | v                       |
| 26   | 1000                    | F800 0000 0000 0000   | 8000 0000 0000          |

| Pass | Address<br>of PTD (C.6) | Complement<br>PTD (C.7) | Complement<br>SVA C.9 |
|------|-------------------------|-------------------------|-----------------------|
| 1    | 1002                    | FFFF FFFF FF80 0000     | FFFF 7FFF FC00        |
| 2    | 1004                    | FFFF FFFF FF40 0000     | FFFF 7FFF FA00        |
| 3    | 1008                    | FFFF FFFF FEC0 0000     | FFFF 7FFF F800        |
| 4    | 1010                    | FFFF FFFF FDC0 0000     | FFFF 7FFF EE00        |
| 5    | 1020                    | FFFF FFFF FBC0 0000     | FFFF 7FFF DE00        |
| 6    | 1040                    | FFFF FFFF F7C0 0000     | FFFF 7FFF BE00        |
| 7    | 1080                    | FFFF FFFF E7C0 0000     | FFFF 7FFF 7E00        |
| 8    | 1100                    | FFFF FFFF D7C0 0000     | FFFF 7FFE FE00        |
| 9    | 1000                    | FFFF FFFF BFC0 0000     | FFFF 7FFD FE00        |
| A    | 1000                    | FFFF FFFF 7FC0 0000     | FFFF 7FFB FE00        |
|      |                         |                         |                       |
| v    | v                       | v                       | v                     |
| 16   | 1000                    | FFFF F7FF FFC0 0000     | FFFF 3FFF FE00        |
| 17   | 1002                    | FFFF EFFF FFC0 0000     | FFFF 7FFF FE00        |
| 18   | 1004                    | FFFF DFFF FFC0 0000     | FFFF 7FFF FE00        |
| 19   | 1008                    | FFFF BFFF FFC0 0000     | FFFF 7FFF FE00        |
| 1A   | 1010                    | FFFF 7FFF FFC0 0000     | FFFF 7FFF FE00        |
| 1B   | 1020                    | FFFF FFFF FFC0 0000     | FFFF 7FFF FE00        |

60000285 A

LMT3/P - 232

| Pass | Address<br>of PTD (C.6) | Complement<br>PTD (C.7) | Complement<br>SVA C.9 |
|------|-------------------------|-------------------------|-----------------------|
| 1C   | 1040                    | FFFD FFFF FFC0 0000     | FFDF 7FFF FE00        |
| 1D   | 1080                    | FFFB FFFF FFC0 0000     | FFBF 7FFF FE00        |
| 1E   | 1100                    | FFF7 FFFF FFC0 0000     | FF7F 7FFF FE00        |
| 1F   | 1000                    | FFEF FFFF FFC0 0000     | FEFF 7FFF FE00        |
| 20   | 1000                    | FFDF FFFF FFC0 0000     | FDFF 7FFF FE00        |
|      |                         |                         |                       |
| v    | v                       | v                       | v                     |
| 26   | 1000                    | F7FF FFFF FFC0 0000     | 7FFF 7FFF FE00        |

| Pass | Expected Results<br>for hits (C.3 and C.8) | Expected Results<br>for misses (C.5 and C.10) |
|------|--------------------------------------------|-----------------------------------------------|
| 1    | 0000 8010 0100 0000                        | 0000 8018 8200 0000                           |
| 2    | 0000 8020 0100 0000                        | 0000 8028 8200 0000                           |
| 3    | 0000 8040 0100 0000                        | 0000 8048 8200 0000                           |
| 4    | 0000 8080 0100 0000                        | 0000 8080 8200 0000                           |
| 5    | 0000 8100 0100 0000                        | 0000 8108 8200 0000                           |
|      |                                            |                                               |
| v    | v                                          | v                                             |
| 16   | 0000 8000 0100 0000                        | 0000 8008 8200 0000                           |
| 17   | 0000 8010 0100 0000                        | 0000 8018 8200 0000                           |
| 18   | 0000 8020 0100 0000                        | 0000 8028 8200 0000                           |
| 19   | 0000 8040 0100 0000                        | 0000 8048 8200 0000                           |
| 1A   | 0000 8080 0100 0000                        | 0000 8088 8200 0000                           |
| 1B   | 0000 8100 0100 0000                        | 0000 8108 8200 0000                           |
|      |                                            |                                               |
| v    | v                                          | v                                             |
| 26   | 0000 8000 0100 0000                        | 0000 8008 8200 0000                           |

Micrands Used:

| Number | Description      |
|--------|------------------|
| 14     | Index Operation. |

Initialization:

- Set CM 0 through 2000(HEX) to 0ACBD3120--0(HEX).
- Set Pass count to 1.

Conditions:

- C.0 Pass count (1 through 26(HEX)).
- Report pass count; informational.

60000285 A

LMT3/P - 233



Conditions:

- C.1 CM address of PTD.
  - a. If Scope Mode or Repeat Iteration set, do not create a new PTD address or PTD. Use the one from the last iteration.
  - b. Report current CM address of PTD; informational.
- C.2 Positive PTD.
  - a. Set CM; address reported in C.1 to current positive PTD.
  - b. Report current positive PTD; informational.
- C.3 Expected result when hit is expected.
  - a. Set RF2 to current positive SVA.
  - b. Report current expected result for hit; informational.
- C.4 Actual result when hit is expected.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not as reported in C.3.
- C.5 Expected result when miss is expected.
  - a. Clear Test Bit in Positive PTD.
  - b. Report current expected result for miss; informational.
- C.6 Actual result when miss is expected.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.5.
- C.7 Complement PTD.
  - a. Set CM; address reported in C.1 to current complement PTD.
  - b. Report current complement PTD; informational.
- C.8 Expected result when hit is expected.
  - a. Set RF2 to current complement SVA.
  - b. Report current expected result for hit; informational.
- C.9 Actual result when hit is expected.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not as reported in C.8.
- C.10 Expected result when miss is expected.
  - a. Set Test Bit in Complement PTD.
  - b. Report current expected result for miss; informational.

Conditions:

- C.11 Actual result when miss is expected.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 14, Index, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - d. Read RF8, report data; error if not as reported in C.10.
- C.12 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.13 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.14 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 28(HEX) passes complete; else advance to next PTD and return to C.0.

## SECTION 22 - RMA MUX, PAGE FRAME ADDRESS ENTRY

This section tests RMA mux entry C. Entry C takes the merged value of the Page Frame Address (PFA) from the PTD in CM and the Page Offset (PO) and submits it to CM as the RMA.

This is the first subsection to perform a complete virtual address operation (CM write).

Subsection 0 - RMA mux, Page Frame Address Entry - Part 1.

Subsection 1 - RMA mux, Page Frame Address Entry - Part 2.

### Method:

To test this entry to the RMA mux a Page Table hit must occur so the PFA can be merged with the PO to form the final RMA.

The write operation is the most basic virtual operation available and it is therefore used in this section.

A Page Table consisting of one entry is set up in CM. The Valid, Used, and Modified bits are set and the Continue bit is clear. The 38-bit Tag is also left clear.

The 22-bit PFA field is the only area in the PTD modified. Various test patterns are set into the PFA and the PO register and a write operation is executed. If the formatted write data is stored at the expected RMA the operation is considered successful.

The test PFA is masked to clear address bits beyond the CM configuration of the system being tested.

CM address 1000(HEX) is always selected for the Page Table Descriptor. This is effected by setting the PTA register to 8, the PSM register to zero, and keeping SVA bits 16 through 47 at zero.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 0.
- Select all Map Sets.
- Select all available Cache Sets.

## Section 22, Subsection 0

### Test RMA mux, Page Frame Address Entry, Part 1.

Part one is the initial ones and zeros test of this mux entry. The Page Frame Address (PFA) from the Page Table Descriptor (PTD) in CM is merged with the Page Offset (PO) to form the final RMA.

### Hardware Tested:

- Input C to RMA mux.
- PFA/PO Merge.
- RMA Mux Control.

### Method:

In this subsection all test values consist of a ones or zeros pattern. These test patterns are set into the SVA (to become the PO) and the PFA portion of a PTD in CM. A write operation is initiated and, if the write data is written in CM at the expected location, the test is successful.

For additional detail, refer to the section description.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| F      | Purge Cache All.   |
| 12     | Purge Map All.     |

### Initialization:

- Purge Cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map all.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).

Conditions:

- C.0 Expected result for 0s test.  
a. Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry, PFA is set to 0.  
b. Set RF2 to 0--0(HEX) (SVA).  
c. Set RFA to C3C3 0000 0000 0000(HEX); write data.  
d. Report write data; informational.
- C.1 Actual result for 0s test.  
a. Set CM(0) to DF--DF(HEX); filler.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM 0, report data; error if not as reported in C.0.
- C.2 Expected result for 1s test.  
a. Set CM 1000(HEX) to B000 0000 0003 FFFF(HEX);+ PTD entry, PFA is set to 1s.  
b. Set RF2 to 0--0FFF8(HEX) (SVA).  
c. Set RFA to C3C3 0000 0000 FFF8(HEX) write data.  
d. Report write data; informational.
- C.3 Actual result for 1's test.  
a. Set CM FFFFFFF(HEX))+ to DF--DF(HEX); filler.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM FFFFFFF(HEX)),+ report data; error if not as reported in C.2.
- C.4 Expected result for 0's/1's test.  
a. Set RF2 to 0--01F8(HEX) (SVA).  
b. Set RFA to C3C3 0000 0000 01F8(HEX); write data.  
c. Report write data; informational.
- C.5 Actual result for 0's/1's test.  
a. Set CM FFFFFFF(HEX))+ to DF--DF(HEX); filler.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM FFFFFFF(HEX)),+ report data; error if not as reported in C.4.
- C.6 Expected result for 1's/0's test.  
a. Set CM 1000(HEX) to B000 0000 0003 FF80(HEX) \*\* (PTD entry, PFA is set to 0 in PO area).  
b. Set RF2 to 0--0FFF8(HEX) (SVA).  
c. Set RFA to C3C3 0000 0000 FFF8(HEX); write data.  
d. Report write data; informational.

\*\*PFA and CM address variables are shown as if testing a system with 128 megabytes of CM. These variables are tailored to fit the configuration under test.

60000285 A

LMT3/P - 238

Conditions:

- C.7 Actual result for 1's/0's test.  
a. Set CM FFFFFFF(HEX))+ to DF--DF(HEX); filler.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM FFFFFFF(HEX)),+ report data; error if not as reported in C.6.
- C.8 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Section 22, Subsection 1

Test RMA MUX, Page Frame Address Entry, Part 2.

Part two continues testing of the PFA(CM)/PO entry to the RMA mux. This subsection also tests the write operation to Cache.

Hardware Tested:

- PFA(CM)/PO Merge.
- Input C to RMA mux.
- Write Cache Gate.

Method:

Operation of this subsection is the same as subsection zero except for patterns (alternate bit patterns) and the Cache setup. Cache is allocated (FAKECM mode) using the same SVA as that was used to perform the write. When the write operation is executed both CM and Cache should be written.

Because extra conditions are required to support the Cache test, two passes are required to exercise all the test patterns.

Test Patterns:

Refer to Conditions.

60000285 A

LMT3/P - 239

# Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read Word from SVA.    |
| E      | Write Word to SVA.     |
| F      | Purge Cache All.       |
| 12     | Purge Map All.         |

## Initialization:

- Purge Cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map all.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select FAKECM mode.

## Conditions:

- C.0 Pass count (1 or 2).  
a. Increment and report pass count; informational.
- C.1 Allocate SVA in Cache.  
a. If pass 1, set RF2 to 0--0AAA8(HEX) (SVA).  
b. If pass 2, set RF2 to 0--05550(HEX) (SVA).  
c. Start processor at micrand C, FAKECM allocate, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Set RFA to DF--DF(HEX); write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Report SVA; informational.
- C.2 Expected result, CM write data.  
a. Deselect FAKECM mode.  
b. If pass 1, set CM 1000(HEX) to B000 0000 0001 5500(HEX);+ PTD entry. If pass 2, set CM 1000(HEX) to B000 0000 0002 AA80(HEX);+ PTD entry.  
c. If pass 1, set RFA to C3C3 0000 0000 AAA8(HEX); write data.  
d. If pass 2, set RFA to C3C3 0000 0000 5550(HEX); write data.  
e. Report write data; informational.

+PFA and CM address variables are shown as if testing a system with 128 megabytes of CM. These variables are tailored to fit the configuration under test.

60000285 A

LMT3/P - 240

# Conditions:

- C.3 Actual result from CM.  
a. Set CM 555555(HEX))+ to DF--DF(HEX); filler, pass 1. Set CM (AAAAA(HEX))\*\* to DF--DF(HEX); filler, pass 2.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. If pass 1, read CM 555555(HEX))+ report data; error if not as reported in C.2.  
e. If pass 2, read CM AAAAAA(HEX))+ report data; error if not as reported in C.2.
- C.4 Actual result from Cache.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Select FAKECM mode.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not as reported in C.2.
- C.5 Allocate SVA in Cache.  
a. If pass 1, set RF2 to 0--0A8(HEX) (SVA).  
b. If pass 2, set RF2 to 0--0150(HEX) (SVA).  
c. Start processor at micrand C, FAKECM allocate, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Set RFA to DF--DF(HEX); write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Report SVA; informational.
- C.6 Expected result, CM write data.  
a. Deselect FAKECM mode.  
b. If pass 1, set CM 1000(HEX) to B000 0000 0001 5555(HEX);+ PTD entry. If pass 2, set CM 1000(HEX) to B000 0000 0002 AAAA(HEX);+ PTD entry.  
c. If pass 1, set RFA to C3C3 0000 0000 00A8(HEX); write data.  
d. If pass 2, set RFA to C3C3 0000 0000 0150(HEX); write data).  
e. Report write data; informational.

\*\*PFA and CM address variables are shown as if testing a system with 128 megabytes of CM. These variables are tailored to fit the configuration under test.

+PFA and CM address variables are shown as if testing a system with 128 megabytes of CM. These variables are tailored to fit the configuration under test.

60000285 A

LMT3/P - 241

Conditions:

- C.7 Actual result from CM.
- Set CM 555555(HEX)+ to DF--DF(HEX); filler, pass 1. Set CM (AAAAA(HEX))+ to DF--DF(HEX) (filler) pass 2.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - If pass 1, read CM 555555(HEX),+ report data; error if not as reported in C.6.
  - If pass 2, read CM AAAAAA(HEX),+ report data; error if not as reported in C.6.
- C.8 Actual result from Cache.
- Set RF6 to DF--DF(HEX); filler.
  - Select FAKECM mode.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not as reported in C.6.
- C.9 Allocate SVA in Cache.
- If pass 1, set RF2 to 0--05550(HEX) (SVA).
  - If pass 2, set RF2 to 0--0AAA8(HEX) (SVA).
  - Start processor at micrand C, FAKECM allocate, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RFA to DF--DF(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.10 Expected result, CM write data.
- Deselect FAKECM mode.
  - If pass 1, set CM 1000(HEX) to B000 0000 0001 5555(HEX);+ PTD entry. If pass 2, set CM 1000(HEX) to B000 0000 0002 AAAA(HEX);+ PTD entry.
  - If pass 1, set RFA to C3C3 0000 0000 5550(HEX); write data.
  - If pass 2, set RFA to C3C3 0000 0000 AAA8(HEX); write data.
  - Report write data; informational.
- C.11 Actual result from CM.
- Set CM 155FEA(HEX)+ to DF--DF(HEX); filler, pass 1. Set CM (ABFD5(HEX))+ to DF--DF(HEX); filler, pass 2.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - If pass 1, read CM 155FEA(HEX),+ report data; error if not as reported in C.10.
  - If pass 2, read CM AABFD5(HEX),+ report data; error if not as reported in C.10.

+ PFA and CM address variables are shown as if testing a system with 128 megabytes of CM. These variables are tailored to fit the configuration under test.

80000285 A

LMT3/P - 242

Conditions:

- C.12 Actual result from Cache.
- Set RF6 to DF--DF(HEX); filler.
  - Select FAKECM mode.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not as reported in C.10.
- C.13 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.15 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if two passes complete; else return to C.0.

Deselect FAKECM mode.

80000285 A

LMT3/P - 243

## SECTION 23 - MAP PFA/PO MERGE INPUT TO FINAL MUX REGISTER

This section tests the OR logic network which merges the Page Frame Address, from the Page Map, with the Page Offset (PO) to form the final RMA. This RMA is selected into the Final RMA Mux register and then sent to CM. This section also tests the Page Map match logic.

Subsection 0 - PFA/PO Merge to Final RMA mux, Sets 0 and 1.

Subsection 1 - PFA/PO Merge to Final RMA mux for Set 2.

Subsection 2 - Page Table Hit, Prevalidate.

### Method:

The following describes the method used in Subsections 0 and 1. The Subsection 2 method is described in Subsection 2.

The PFA(MAP)/PO logic network is tested by setting bit operands into the Page Frame Address field of the Page Map, then issuing a CM write using a SVA to match the Map entry. This causes the operand value in the PFA field to be merged with the operand value from the PO, creating the final RMA directing where the write data is to be stored. To do this, the following steps are executed.

1. All Page Map Sets are purged.
2. A Page Table Entry is set up in CM 1000(HEX) with the Valid, Used, and Modified bits set and the Continue bit clear. The 38-bit Tag is clear and the 22-bit PFA is set to a bit pattern.
3. A CM Write is issued using a SVA of zero. A Map miss occurs, a Page Table Hit occurs and the formatted write data is written into CM at the computed RMA. Also the Page Map is entered with the bit pattern from the PFA.
4. The PTD is now cleared in CM. This eliminates the possibility of a Page Table Hit during the Test Operation; see Micrand 15 description.
5. The data written to CM by step 3 is cleared.
6. The Test Operation is initiated, causing a Page Map Hit. The PFA field is merged with the PO to form the RMA, which is then sent to the Register File on the RDATA lines.

The Test Operation is used to convert the incoming SVA to the correct RMA. Page Map Sets 0 and 1 are used in subsection 0 and Set 2 in subsection 1.

A set of operands is used for each Map Set during each pass.

Five passes are required to exercise all operands.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 0.
- Deselect all Cache Sets.
- Set CM 0 through 2000(HEX) to 0ACBD3120--0(HEX).

## Section 23, Subsection 0

Test PFA/PO Merge to Final RMA mux, Sets 0 and 1.

This subsection tests the OR logic network that merges the PFA (Map) and the PO to form the final RMA. This subsection uses Map Sets 0 and 1.

### Hardware Tested:

- PFA(Map)/PO Merge.
- Final RMA Mux register.
- PFA(Map) Output.
- Page Map Match.
- Test Operation Control.

### Method:

Ten combinations of PFA/PO are required to test the OR network. In order to enhance hardware isolation, two patterns are used with two sets on each pass. For example:

Conditions 1 through 3 use Map set 0 with pattern 1.

Conditions 4 through 6 use Map set 0 with pattern 2.

Conditions 7 through 9 use Map set 1 with pattern 1.

Conditions 10 through 12 use Map set 1 with pattern 2.

For additional detail, see the section description.

# Test Patterns (hexadecimal):

| Pass | Condition | PFA+  | PO++ | RMA+   | Expected Results+   |
|------|-----------|-------|------|--------|---------------------|
| 1    | 3,9       | 00000 | 0000 | 000000 | 0000 0000 0000 0000 |
| 1    | 6,12      | 3FFFF | FFF8 | FFFFFF | 07FF FFF8 0000 0000 |
| 2    | 3,9       | 3FF80 | FFF8 | FFFFFF | 07FF FFF8 0000 0000 |
| 2    | 6,12      | 3FFFF | 01F8 | FFFFFF | 07FF FFF8 0000 0000 |
| 3    | 3,9       | 15555 | 00A8 | 555555 | 02AA AAA8 0000 0000 |
| 3    | 6,12      | 15500 | AAA8 | 555555 | 02AA AAA8 0000 0000 |
| 4    | 3,9       | 2AAAA | 0150 | AAAAAA | 0155 5550 0000 0000 |
| 4    | 6,12      | 2AA80 | 5550 | AAAAAA | 0155 5550 0000 0000 |
| 5    | 3,9       | 15555 | 5550 | 555FEA | 02AA FF50 0000 0000 |
| 5    | 6,12      | 2AAAA | AAA8 | AABFD5 | 0155 FEA8 0000 0000 |

+The values shown are the values used with a 128 megabyte CM configuration, the actual value used would be masked to reflect the CM configured with the system under test.

++The PO value shown is really the low order 16 bits of the SVA from which the PO is derived.

Data written to CM is formatted as C3C3 xxxx C3C3 yyyy(HEX).

xxxx = Current PFA (lower 16 bits only).  
yyyy = Current PO.

## Micrands Used:

| Number | Description         |
|--------|---------------------|
| E      | Write Word to SVA.  |
| 12     | Purge Page Map All. |
| 15     | Test Operation.     |

## Initialization:

- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 5).  
a. Report pass count; informational.

## Conditions:

- C.1 Write data to CM, initialize Page Map, report write data.  
a. Start processor at micrand 12, purge map all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Select Page Map Set 0.  
d. Set CM 1000(HEX) to B000 0000 000x xxxx(HEX); xxxxx(HEX) is the current PFA.  
e. Set RF2 to 0000 0000 yyyy(HEX) (SVA); yyyy is the current PO.  
f. Set RFA to current write data.  
g. Start processor at micrand E, write word to SVA, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Read word from CM at current RMA, report data; error if not as formatted in RFA.
- C.2 Expected result from test operation.  
a. Set CM 1000(HEX) to 0; clear PTD.  
b. Set CM at current RMA to DF--DF(HEX); filler.  
c. Report current expected result; informational.
- C.3 Actual result from test operation.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.2.
- C.4 Write data to CM, initialize Page Map, report write data.  
a. Start processor at micrand 12, purge map all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set CM 1000(HEX) to B000 0000 000x xxxx(HEX); xxxxx(HEX) is the current PFA.  
d. Set RF2 to 0000 0000 yyyy(HEX) (SVA); yyyy is the current PO.  
e. Set RFA to current write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Read word from CM at current RMA, report data; error if not as formatted in RFA.
- C.5 Expected result from test operation.  
a. Set CM 1000(HEX) to 0; clear PTD.  
b. Set CM at current RMA to DF--DF(HEX); filler.  
c. Report current expected result; informational.
- C.6 Actual result from test operation.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as reported in C.5.

Conditions:

- C.7 Write data to CM, initialize Page Map, report write data.  
 a. Start processor at micrand 12, purge map all, wait for CPU halted.  
 b. Read CPU status, format and save if error; error if not 0.  
 c. Select Page Map Set 1.  
 d. Set CM 1000(HEX) to B000 0000 000x xxxx(HEX); xxxxx(HEX) is the current PFA.  
 e. Set RF2 to 0000 0000 yyyy(HEX) (SVA); yyyy is the current PO.  
 f. Set RFA to current write data.  
 g. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 h. Read CPU status, format and save if error; error if not 0.  
 i. Read word from CM at current RMA, report data; error if not as formatted in RFA.
- C.8 Expected result from test operation.  
 a. Set CM 1000(HEX) to 0; clear PTD.  
 b. Set CM at current RMA to DF--DF(HEX); filler.  
 c. Report current expected result; informational.
- C.9 Actual result from test operation.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand 15, Test, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not as reported in C.8.
- C.10 Write data to CM, initialize Page Map, report write data.  
 a. Start processor at micrand 12, purge map all, wait for CPU halted.  
 b. Read CPU status, format and save if error; error if not 0.  
 c. Set CM 1000(HEX) to B000 0000 000x xxxx(HEX); xxxxx(HEX) is the current PFA.  
 d. Set RF2 to 0000 0000 yyyy(HEX) (SVA); yyyy is the current PO.  
 e. Set RFA to current write data.  
 f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 g. Read CPU status, format and save if error; error if not 0.  
 h. Read word from CM at current RMA, report data; error if not as formatted in RFA.
- C.11 Expected result from test operation.  
 a. Set CM 1000(HEX) to 0; clear PTD.  
 b. Set CM at current RMA to DF--DF(HEX); filler.  
 c. Report current expected result; informational.
- C.12 Actual result from test operation.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand 15, Test, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not as reported in C.11.
- C.13 CPU Status Word 1.  
 a. Report word 1 if saved; error if reported.

Conditions:

- C.14 CPU Status Word 2.  
 a. Report word 2 if saved; error if reported.
- C.15 First Failure Capture Information.  
 a. Report PFS87 bits 19-23 in bits 59-63; information only.  
 b. Report last PTL written in bits 34-47; information only.  
 c. Report last PTA written in bits 13-31; information only.

Exit Subsection if five passes complete; else advance current operand pointer and return to C.0.

Section 23, Subsection 1

Test PFA/PO Merge to Final RMA mux for Set 2.

This subsection tests the OR logic network that merges the PFA (Map) and the PO to form the final RMA. This subsection uses Map Set 2.

Hardware Tested:

- PFA(Map)/PO Merge.
- Final RMA Mux register.
- PFA(Map) Output.
- Page Map Match.
- Test Operation Control.

Method:

Ten combinations of PFA/PO are required to test the OR network. In order to enhance hardware isolation, two patterns are used for set two on each pass. For example:

Conditions 1 through 3 use Map set 2 with pattern 1.

Conditions 4 through 6 use Map set 2 with pattern 2.

For additional detail, see the section description.



# Test Patterns (hexadecimal):

| Pass | Condition | PFA+  | PO++ | RMA+   | Expected Results+   |
|------|-----------|-------|------|--------|---------------------|
| 1    | 3         | 0000  | 0000 | 000000 | 0000 0000 0000 0000 |
| 1    | 6         | 3FFFF | FFF8 | FFFFFF | 07FF FFF8 0000 0000 |
| 2    | 3         | 3FF80 | FFF8 | FFFFFF | 07FF FFF8 0000 0000 |
| 2    | 6         | 3FFFF | 01F8 | FFFFFF | 07FF FFF8 0000 0000 |
| 3    | 3         | 15555 | 00A8 | 555555 | 02AA AAA8 0000 0000 |
| 3    | 6         | 15500 | AAA8 | 555555 | 02AA AAA8 0000 0000 |
| 4    | 3         | 2AAAA | 0150 | AAAAAA | 0555 5550 0000 0000 |
| 4    | 6         | 2AA80 | 5550 | AAAAAA | 0555 5550 0000 0000 |
| 5    | 3         | 15555 | 5550 | 555FEA | 02AA FF50 0000 0000 |
| 5    | 6         | 2AAAA | AAA8 | AABFD5 | 0555 FEA8 0000 0000 |

+The values shown are the values used with a 128 megabyte CM configuration.  
The actual value used would be masked to reflect the CM configured with the system under test.

++The PO value shown is really the low order 16 bits of the SVA from which the PO is derived.

Data written to CM is formatted as C3C3 xxxx C3C3 yyyy(HEX).

xxxx = Current PFA. (Lower 16 bits).  
yyyy = Current PO.

## Micrands Used:

| Number | Description         |
|--------|---------------------|
| E      | Write Word to SVA.  |
| 12     | Purge Page Map All. |
| 15     | Test Operation.     |

## Initialization:

- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 5).  
a. Report pass count; informational.

## Conditions:

- C.1 Write data to CM, initialize Page Map, report write data.  
a. Start processor at micrand 12, purge map all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Select Page Map Set 2.  
d. Set CM 1000(HEX) to B000 0000 000x xxxx(HEX); xxxx(HEX) is the current PFA.  
e. Set RF2 to 0000 0000 yyyy(HEX) (SVA); yyyy is the current PO.  
f. Set RFA to current write data.  
g. Start processor at micrand E, write word to SVA, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Read word from CM at current RMA, report data; error if not as formatted in RFA.
- C.2 Expected result from test operation.  
a. Set CM 1000(HEX) to 0; clear PTD.  
b. Set CM at current RMA to DF--DF(HEX); filler.  
c. Report current expected result; informational.
- C.3 Actual result from test operation.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not as reported in C.2.
- C.4 Write data to CM, initialize Page Map, report write data.  
a. Start processor at micrand 12, purge map all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set CM 1000(HEX) to B000 0000 000x xxxx(HEX); xxxx(HEX) is the current PFA.  
d. Set RF2 to 0000 0000 yyyy(HEX) (SVA); yyyy is the current PO.  
e. Set RFA to current write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Read word from CM at current RMA, report data; error if not as formatted in RFA.
- C.5 Expected result from test operation.  
a. Set CM 1000(HEX) to 0; clear PTD.  
b. Set CM at current RMA to DF--DF(HEX); filler.  
c. Report current expected result; informational.
- C.6 Actual result from Test operation.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not as reported in C.5.
- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

Conditions:

- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if five passes complete; else advance current operand pointer and return to C.0.

Section 23, Subsection 2

This subsection tests the Page Table Hit Enable Gate. This gate enables a Page Table Hit when the Valid bit is set in the Page Table Descriptor (PTD), or during a Prevalidated operation, or during an Index operation.

Hardware Tested:

- Page Table Hit Enable.
- Prevalidated Control.

Method:

In previous sections the Valid entry to the Page Table Enable Gate was tested. This subsection now tests the remaining two entries.

Method:

Four operations are used to ensure all gate entries work. They are:

- C.0 A Page Table Hit using the Valid bit. This has been tested but is used here to provide continuity for the next tests.
- C.3 A Page Table Hit during an Index operation. The Valid bit in the PTD is left clear. Up till now the Valid bit has been set with Index Operations.
- C.6 A Prevalidated Page Table Hit. Again the Valid bit is clear. A Byte Store operation is used to test the Prevalidated signal. The Byte Store is automatically preceded by a prevalidate request from I1. This will result in a Page Table miss (valid bit clear) and the PSWF status is set into the MCR. However, because the Word Valid bit is set in the ACMIC portion of the Byte Store micrand, the valid bit in the Descriptor is ignored and the operation is completed. The fact that the write completes proves the gate is functioning correctly.

Conditions:

- C.9 An operation with none of the three above situations. This should result in a miss.

Test Patterns (hexadecimal):

| Condition | PTD                 | SVA            |
|-----------|---------------------|----------------|
| 0         | B000 0000 0000 0000 | 0000 0000 0008 |
| 3         | 3000 0000 0000 0001 | 0000 0000 0008 |
| 6         | 3000 0000 0000 0001 | 0000 0000 0008 |
| 9         | 3000 0000 0000 0001 | 0000 0000 0008 |

Micrands Used:

| Number | Description                                                                          |
|--------|--------------------------------------------------------------------------------------|
| E      | Write Word to SVA.                                                                   |
| 12     | Purge Map All.                                                                       |
| 14     | Index Operation.                                                                     |
| 18     | Prevalidated Operation (same as Byte Store to SVA with Word Valid bit set in ACMIC). |

Initialization:

- Deselect all Map Sets.
- Set RF12 to 0--07; byte count -1.
- Set PSM to 7F(HEX).
- Set CM; 0 through FFF(HEX) to C3C3 C3C3 C300 0xxx(HEX); where xxx (HEX) = 0 through FFF(HEX).

Conditions:

- C.0 Purge Page Map, execute Valid Bit Test.  
a. Start processor at micrand 12, purge map all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set RF2 to 0--080.  
d. Set RFA to F7--F7(HEX); write data.  
e. Set CM 1000(HEX) to B000 0000 0000 0000(HEX) (PTD).  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CM 1, report data; error if not F7--F7(HEX).
- C.1 CPU Status Word 1.  
a. Read CPU status, format and report word one; error if not 0.

Conditions:

- C.2 CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.3 Index Hit Test.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Set CM 1000(HEX) to 3000 0000 0000 0001(HEX) (PTD).
  - c. Start processor at micrand 14, Index, wait for CPU halted.
  - d. Read RF8, report data; error if not 0000 8000 0100 0000(HEX).
- C.4 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.5 CPU Status Word 2.
  - a. Report word 2; error if not 0.
- C.6 Prevalidate Hit Test.
  - a. Set CM 41(HEX) to C3--C3000041(HEX).
  - b. Start processor at micrand 16, Prevalidated, wait for CPU halted.
  - c. Read CM 41(HEX), report data; error if not F7--F7(HEX).
- C.7 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.8 CPU Status Word 2.
  - a. Report word 2; error if not 0000040--0(HEX) (PSWF).
- C.9 Page Table Miss Test.
  - a. Set CM 41(HEX) to C3--C3000041(HEX).
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CM 41(HEX), report data; error if not C3--C3000041(HEX); write not permitted due to miss.
- C.10 CPU Status Word 1.
  - a. Read CPU status, format and report word 1; error if not 0.
- C.11 CPU Status Word 2.
  - a. Report word 2; error if not 0000 040--0(HEX) (PSWF).
- C.12 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

60000285 A

LMT3/P - 254

SECTION 24 - PAGE MAP DATA INTEGRITY

This section tests the Page Map data RAMs for data integrity. All basic patterns, 0s, Fs; and alternate bit patterns, are used in all three sets.

Subsection 0 - Page Map Data Test, Zeros Pattern.

Subsection 1 - Page Map Data Test, Ones Pattern.

Subsection 2 - Page Map Data Test, Alternate Bits 1.

Subsection 3 - Page Map Data Test, Alternate Bits 2.

Method:

The Page Maps cannot be directly written or read, so they must be data checked using an implicit write and read.

The write is performed by setting a pattern in a Page Table Descriptor (PTD) in CM and issuing a write CM sequence, resulting in a Page Table Hit and a Page Map Allocate and write.

The read is performed when a Test Operation is issued. A Page Map Hit occurs and the resulting RMA is transmitted to the Register File.

The 33-bit Page Map Tag and the 22-bit\*\* Page Frame Address (PFA) are set to the current test pattern.

Bits 50 through 54 of the SVA represent the Page Map Address. This field is incremented from 0 through 1F(HEX) to test all 32 locations.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).

Section 24, Subsection 0

This subsection tests the data integrity of all three Page Map Sets using a zeros pattern.

Hardware Tested:

- Page Map RAMs.
- SPID Fan-out to Page Map.

\*\*The PFA is masked to reflect the CM configuration of the system under test.

60000285 A

LMT3/P - 255

# Method:

Conditions 1 and 2 comprise the write loop. All Page Map locations in the selected set are allocated and written. If any error is sensed the write loop is aborted.

Conditions 3 through 10 make up the read loop. Four locations are read during each pass. Therefore 8 passes through the read loop are required. If any error is detected during the read loop the subsection is exited.

A Page Table is set up at CM 1000(HEX), 32 locations long. This is used during the write loop to provide needed PTDs to match the incoming SVAs.

## Test Patterns (hexadecimal):

| SVAs           | CM Address<br>of PTD | PTD                 |
|----------------|----------------------|---------------------|
| 0000 0000 0000 | 001000               | B000 0000 0000 0000 |
| 0000 0000 0200 | 001002               | B000 0000 0040 0000 |
| 0000 0000 0400 | 001004               | B000 0000 0080 0000 |
| 0000 0000 0600 | 001006               | B000 0000 00C0 0000 |
| 0000 0000 0800 | 001008               | B000 0000 0100 0000 |
|                |                      |                     |
| 0000 0000 3E00 | 00103E               | B000 0000 07C0 0000 |

Data written to CM is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = Current Map set.  
yyyy = Current SVA.

Expected result for all Test Operations is zero.

## Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 15     | Test Operation.    |
| 12     | Purge Map All.     |

## Initialization:

- Set up Page Table at CM; 1000(HEX) through 103E(HEX).
- Set CM; 0 through FFF(HEX) to C3--C30xxx(HEX) (xxx = 0 through FFF(HEX)).
- Set current Page Map set to 0.

## Conditions:

- C.0 Current Set being tested (0 through 2).
- Select current Map Set.
  - Start processor at micrand 12, purge map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set current SVA to 0.
  - Report current set; informational.
- C.1 Write to SVA, report SVA.
- Set RF2 to current SVA.
  - Set RFA to current write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.2 Report data written to CM.
- Read CM 0, report data; error if not as written in C.1.
- Continue to next condition if error, SSM, or 32 passes complete; else advance SVA (+200(HEX) and return to C.1.
- Set current SVA to zero.
- C.3 Report current SVA.
- Set RF2 to current SVA.
  - Report current SVA; informational.
- C.4 Test operation result.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0000 0000 0100 0000(HEX).
- C.5 Report current SVA.
- Advance current SVA (+200(HEX)).
  - Set RF2 to current SVA.
  - Report current SVA; informational.
- C.6 Test operation result.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0000 0000 0100 0000(HEX).
- C.7 Report current SVA.
- Advance current SVA (+200(HEX)).
  - Set RF2 to current SVA.
  - Report current SVA; informational.

# Conditions:

- C.8 Test operation result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not 0000 0000 0100 0000(HEX).
- C.9 Report current SVA.  
a. Advance current SVA (+200(HEX)).  
b. Set RF2 to current SVA.  
c. Report current SVA; informational.
- C.10 Test operation result.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not 0000 0000 0100 0000(HEX).
- C.11 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.  
b. Advance current SVA (+200(HEX)).  
c. If no error and current SVA is not 0--04000(HEX), return to C.3.  
d. If no error and current set is not 3, advance set number and return to C.0.
- C.13 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit subsection if set 2 completely tested; else advance SVA (200(HEX) and if not 4000(HEX), return to C.3; else return to C.0.

## Section 24, Subsection 1

This subsection tests the data integrity of all three Page Map Sets using a ones pattern.

## Hardware Tested:

- Page Map RAMs.
- SPID Fan-out to Page Map.

Conditions 1 and 2 comprise the write loop. All Page Map locations in the selected set are allocated and written. If any error is sensed the write loop is aborted.

80000285 A

LMT3/P - 258

Conditions 3 through 10 make up the read loop. Four locations are read during each pass. Therefore 8 passes through the read loop are required. If any error is detected during the read loop the subsection is exited.

A Page Table is set up at CM 1000(HEX), 32 locations long. This is used during the write loop to provide needed PTDs to match the incoming SVAs.

## Test Patterns (hexadecimal):

| SVAs           | CM Address<br>of PTD | PTD                  |
|----------------|----------------------|----------------------|
| FFFF 7FFF FE00 | 001000               | BFFF FFFF FFC3 FFFF+ |
| FFFF 7FFF FC00 | 001002               | BFFF FFFF FF83 FFFF  |
| FFFF 7FFF FA00 | 001004               | BFFF FFFF FF43 FFFF  |
| FFFF 7FFF F800 | 001006               | BFFF FFFF FF03 FFFF  |
| FFFF 7FFF F600 | 001008               | BFFF FFFF FEC3 FFFF  |
| ↓              | ↓                    | ↓                    |
| FFFF 7FFF C000 | 00103E               | BFFF FFFF F803 FFFF  |

Data written to CM is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = Current Map set.  
yyyy = Current SVA.

\*\*Expected result for all Test Operations is 0xFF FE00 0000 0000(HEX) where;

x = 0 for memory size equal to or less than 16 MB.  
x = 1 for memory size of 32 MB.  
x = 3 for memory size of 64 MB.  
x = 7 for memory size of 128 MB.  
etc.

## Micrands Used:

| Number | Description                       |
|--------|-----------------------------------|
| E      | Write Word to SVA.                |
| 15     | Test Operation 12, Purge Map All. |

## Initialization:

- Set up Page Table at CM; 1000(HEX) through 103E(HEX).
- Set CM; 0 through FFF(HEX) to C3--C30xxx(HEX) (xxx = 0 through FFF(HEX)).
- Set current Page Map set to 0.

80000285 A

LMT3/P - 259

Conditions:

- C.0 Current Set being tested (0 through 2).
- Select current Map Set.
  - Start processor at micrand 12, purge map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set current SVA to FFFF 7FFF FE00(HEX).
  - Report current set; informational.
- C.1 Write to SVA, report SVA.
- Set RF2 to current SVA.
  - Set RFA to current write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.

- C.2 Report data written to CM.
- Read CM FFFFC0(HEX),+ report data; error if not as written in C.1.

Continue to next condition if error, SSM, or 32 passes complete; else reduce SVA (-200(HEX) and return to C.1.

Set CM FFFFC0(HEX) to C3--C3FFFFC0(HEX).\*\*

Set current SVA to FFFF 7FFF FE00(HEX).

- C.3 Report current SVA.
- Set RF2 to current SVA.
  - Report current SVA; informational.
- C.4 Test operation result.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0xFF FE00 0100 0000(HEX).\*\*
- C.5 Report current SVA.
- Reduce current SVA (-200(HEX).
  - Set RF2 to current SVA.
  - Report current SVA; informational.
- C.6 Test operation result.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0xFF FE00 0100 0000(HEX).\*\*

\*\*Variables are shown configured to a 128 megabyte system. The PFA will be masked to support the CM configuration of the system being tested.

60000285 A

LMT3/P - 260

Conditions:

- C.7 Report current SVA.
- Reduce current SVA (-200(HEX).
  - Set RF2 to current SVA.
  - Report current SVA; informational.
- C.8 Test operation result.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0xFF FE00 0100 0000(HEX).\*\*
- C.9 Report current SVA.
- Reduce current SVA (-200(HEX).
  - Set RF2 to current SVA.
  - Report current SVA; informational.
- C.10 Test operation result.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0xFF FE00 0100 0000(HEX).\*\*
- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Reduce current SVA (-200(HEX).

If no error and current SVA is not 0--0BE00(HEX), return to C.3.

If no error and current set is not 2, advance set number and return to C.0.

Section 24, Subsection 2

This subsection tests the data integrity of all three Page Map Sets using an alternate bit pattern.

Hardware Tested:

- Page Map RAMs.
- SPID Fan-out to Page Map.

60000285 A

LMT3/P - 261

# Method:

Conditions 1 and 2 comprise the write loop. All Page Map locations in the selected set are allocated and written. If any error is sensed the write loop is aborted.

Conditions 3 through 10 make up the read loop. Four locations are read during each pass. Therefore 8 passes through the read loop are required. If any error is detected during the read loop the subsection is exited.

A Page Table is set up at CM 1000(HEX), 32 locations long. This is used during the write loop to provide needed PTDs to match the incoming SVAs.

## Test Patterns (hexadecimal):

| SVAs           | CM Address<br>of PTD | PTD                   |
|----------------|----------------------|-----------------------|
| 5555 2AAA 8000 | 102A                 | 8555 5555 5001 5555** |
| 5555 2AAA 8200 | 1028                 | 8555 5555 5041 5555** |
| 5555 2AAA 8400 | 102E                 | 8555 5555 5081 5555** |
| 5555 2AAA 8600 | 102C                 | 8555 5555 50C1 5555** |
| 5555 2AAA 8800 | 1022                 | 8555 5555 5101 5555** |
| 5555 2AAA 8A00 | 1020                 | 8555 5555 5141 5555** |
| 5555 2AAA 8C00 | 1026                 | 8555 5555 5181 5555** |
| 5555 2AAA 8E00 | 1024                 | 8555 5555 51C1 5555** |
| 5555 2AAA 9000 | 103A                 | 8555 5555 5201 5555** |
| 5555 2AAA 9200 | 1038                 | 8555 5555 5241 5555** |
| 5555 2AAA 9400 | 103E                 | 8555 5555 5281 5555** |
| 5555 2AAA 9600 | 103C                 | 8555 5555 52C1 5555** |
| 5555 2AAA 9800 | 1032                 | 8555 5555 5301 5555** |
| 5555 2AAA 9A00 | 1030                 | 8555 5555 5341 5555** |
| 5555 2AAA 9C00 | 1036                 | 8555 5555 5381 5555** |
| 5555 2AAA 9E00 | 1034                 | 8555 5555 53C1 5555** |
| 5555 2AAA A000 | 100A                 | 8555 5555 5401 5555** |
| 5555 2AAA A200 | 1008                 | 8555 5555 5441 5555** |
| 5555 2AAA A400 | 100E                 | 8555 5555 5481 5555** |
| 5555 2AAA A600 | 100C                 | 8555 5555 54C1 5555** |
| 5555 2AAA A800 | 1002                 | 8555 5555 5501 5555** |
| 5555 2AAA AA00 | 1000                 | 8555 5555 5541 5555** |
| 5555 2AAA AC00 | 1006                 | 8555 5555 5581 5555** |
| 5555 2AAA AE00 | 1004                 | 8555 5555 55C1 5555** |
| 5555 2AAA B000 | 101A                 | 8555 5555 5601 5555** |
| 5555 2AAA B200 | 1018                 | 8555 5555 5641 5555** |
| 5555 2AAA B400 | 101E                 | 8555 5555 5681 5555** |
| 5555 2AAA B600 | 101C                 | 8555 5555 56C1 5555** |
| 5555 2AAA B800 | 1012                 | 8555 5555 5701 5555** |
| 5555 2AAA BA00 | 1010                 | 8555 5555 5741 5555** |
| 5555 2AAA BC00 | 1016                 | 8555 5555 5781 5555** |
| 5555 2AAA BE00 | 1014                 | 8555 5555 57C1 5555** |

\*\*Variables are shown configured to a 128 megabyte system. The PFA will be masked to support the CM configuration of the system being tested.

80000285 A

LMT3/P - 282

Data written to CM is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = Current MAP set.  
yyyy = Current SVA.

Expected result for all Test Operations is 02AA AA00 0000 0000(HEX).\*\*

## Micrands Used:

| Number | Description                       |
|--------|-----------------------------------|
| E      | Write Word to SVA.                |
| 15     | Test Operation 12, Purge Map All. |

## Initialization:

- Set up Page Table at CM; 1000(HEX) through 103E(HEX).
- Set CM; 0 through FFF(HEX) to C3--C30xxx(HEX) (xxx = 0 through FFF (HEX)).
- Set current Page Map set to 0.

## Conditions:

- C.0 Current Set being tested (0 through 2).
- a. Select current MAP set.
  - b. Start processor at micrand 12, purge map all, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Set current SVA to 5555 2AAA 8000(HEX).
  - e. Report current set; informational.
- C.1 Write to SVA, report SVA.
- a. Set RF2 to current SVA.
  - b. Set RFA to current write data.
  - c. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report current SVA; informational.
- C.2 Report data written to CM.
- a. Read CM 0, report data; error if not as written in C.1.

Continue to next condition if error, SSM, or 32 passes complete; else advance SVA (+200(HEX)) and return to C.1.

Set current SVA to 5555 2AAA 8000(HEX).

- C.3 Report current SVA.
- a. Set RF2 to current SVA.
  - b. Report current SVA; informational.

80000285 A

LMT3/P - 283

# Conditions:

- C.4 Test operation result.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand 15, Test, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not 02AA AA00 0100 0000(HEX).\*\*
- C.5 Report current SVA.
  - a. Advance current SVA (+200(HEX).
  - b. Set RF2 to current SVA.
  - c. Report current SVA; informational.
- C.6 Test operation result.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand 15, Test, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not 02AA AA00 0100 0000(HEX).\*\*
- C.7 Report current SVA.
  - a. Advance current SVA (+200(HEX).
  - b. Set RF2 to current SVA.
  - c. Report current SVA; informational.
- C.8 Test operation result.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand 15, Test, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not 02AA AA00 0100 0000(HEX).\*\*
- C.9 Report current SVA.
  - a. Advance current SVA (+200(HEX).
  - b. Set RF2 to current SVA.
  - c. Report current SVA; informational.
- C.10 Test operation result.
  - a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand 15, Test, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not 02AA AA00 0100 0000(HEX).\*\*
- C.11 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

\*\*Variables are shown configured to a 128 megabyte system. The PFA will be masked to support the CM configuration of the system being tested.

60000285 A

LMT3/P - 264

Advance current SVA (+200(HEX).

If no error and current SVA is not 0--0BE00(HEX), return to C.3.

If no error and current set is not 2, advance set number and return to C.0.

## Section 24, Subsection 3

This subsection tests the data integrity of all three Page Map Sets using an alternate bit pattern.

## Hardware Tested:

- Page Map RAMs.
- SPID Fan-out to Page Map.

## Method:

Conditions 1 and 2 comprise the write loop. All Page Map locations in the selected set are allocated and written. If any error is sensed the write loop is aborted.

Conditions 3 through 10 make up the read loop. Four locations are read during each pass. Therefore 8 passes through the read loop are required. If any error is detected during the read loop the subsection is exited.

A Page Table is set up at CM 1000(HEX), 32 locations long. This is used during the write loop to provide needed PTDs to match the incoming SVA's.

## Test Patterns (hexadecimal):

| SVA's          | CM Address<br>of PTD | PTD                   |
|----------------|----------------------|-----------------------|
| AAAA 5555 4000 | 1014                 | BAAA AAAA A802 AAAA** |
| AAAA 5555 4200 | 1018                 | BAAA AAAA A842 AAAA** |
| AAAA 5555 4400 | 1010                 | BAAA AAAA A882 AAAA** |
| AAAA 5555 4600 | 1012                 | BAAA AAAA A8C2 AAAA** |
| AAAA 5555 4800 | 101C                 | BAAA AAAA A902 AAAA** |
| AAAA 5555 4A00 | 101E                 | BAAA AAAA A942 AAAA** |
| AAAA 5555 4C00 | 1018                 | BAAA AAAA A982 AAAA** |
| AAAA 5555 4E00 | 101A                 | BAAA AAAA A9C2 AAAA** |
| AAAA 5555 5000 | 1004                 | BAAA AAAA AA02 AAAA** |
| AAAA 5555 5200 | 1006                 | BAAA AAAA AA42 AAAA** |
| AAAA 5555 5400 | 1000                 | BAAA AAAA AA82 AAAA** |
| AAAA 5555 5600 | 1002                 | BAAA AAAA AAC2 AAAA** |
| AAAA 5555 5800 | 100C                 | BAAA AAAA AB02 AAAA** |
| AAAA 5555 5A00 | 100E                 | BAAA AAAA AB42 AAAA** |
| AAAA 5555 5C00 | 1008                 | BAAA AAAA AB82 AAAA** |
| AAAA 5555 5E00 | 100A                 | BAAA AAAA ABC2 AAAA** |
| AAAA 5555 6000 | 1034                 | BAAA AAAA AC02 AAAA** |

60000285 A

LMT3/P - 265



| SVAs           | CM Address<br>of PTD | PTD                   |
|----------------|----------------------|-----------------------|
| AAAA 5555 8200 | 1036                 | BAAA AAAA AC42 AAAA** |
| AAAA 5555 8400 | 1030                 | BAAA AAAA AC82 AAAA** |
| AAAA 5555 8600 | 1032                 | BAAA AAAA ACC2 AAAA** |
| AAAA 5555 8800 | 103C                 | BAAA AAAA AD02 AAAA** |
| AAAA 5555 8A00 | 103E                 | BAAA AAAA AD42 AAAA** |
| AAAA 5555 8C00 | 1038                 | BAAA AAAA AD82 AAAA** |
| AAAA 5555 8E00 | 103A                 | BAAA AAAA ADC2 AAAA** |
| AAAA 5555 9000 | 1024                 | BAAA AAAA AE02 AAAA** |
| AAAA 5555 9200 | 1026                 | BAAA AAAA AE42 AAAA** |
| AAAA 5555 9400 | 1020                 | BAAA AAAA AE82 AAAA** |
| AAAA 5555 9600 | 1022                 | BAAA AAAA AEC2 AAAA** |
| AAAA 5555 9800 | 102C                 | BAAA AAAA AF02 AAAA** |
| AAAA 5555 9A00 | 102E                 | BAAA AAAA AF42 AAAA** |
| AAAA 5555 9C00 | 1028                 | BAAA AAAA AF82 AAAA** |
| AAAA 5555 9E00 | 102A                 | BAAA AAAA AFC2 AAAA** |

Data written to CM is formatted as: C3xx yyyy yyyy(HEX).

xx = Current Map set.  
yyyy = Current SVA.

Expected result for all Test Operations is 0555 5400 0000 0000(HEX).\*\*

\*\*Variables are shown configured to a 128 megabyte system. The PFA will be masked to support the CM configuration of the system being tested.

#### Micrands Used:

| Number | Description                       |
|--------|-----------------------------------|
| E      | Write Word to SVA                 |
| 15     | Test Operation 12, Purge Map All. |

#### Initialization:

- Set up Page Table at CM; 1000(HEX) through 103E(HEX).
- Set CM; (0 through FFF(HEX) to C3--C30xx(HEX) (xxx = 0 through FFF(HEX)).
- Set current Page Map set to 0.

#### Conditions:

- C.0 Current Set being tested; 0 through 2.
- Select current Map set.
  - Start processor at micrand 12, purge map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set current SVA to AAAA 5555 4000(HEX).
  - Report current set; informational.

80000285 A

LMT3/P - 266

#### Conditions:

- C.1 Write to SVA, report SVA.
- Set RF2 to current SVA.
  - Set RFA to current write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report current SVA; informational.
- C.2 Report data written to CM.
- Read CM 0, report data; error if not as written in C.1.
- Continue to next condition if error, SSM, or 32 passes complete; else advance SVA (+200(HEX) and return to C.1.
- Set current SVA to AAAA 5555 4000(HEX).
- C.3 Report current SVA.
- Set RF2 to current SVA.
  - Report current SVA; informational.
- C.4 Test operation result.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0555 5400 0100 0000(HEX).\*\*
- C.5 Report current SVA.
- Advance current SVA (+200(HEX).
  - Set RF2 to current SVA.
  - Report current SVA; informational.
- C.6 Test operation result.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0555 5400 0100 0000(HEX).\*\*
- C.7 Report current SVA.
- Advance current SVA (+200(HEX).
  - Set RF2 to current SVA.
  - Report current SVA; informational.
- C.8 Test operation result.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0555 5400 0100 0000(HEX).\*\*

\*\*Variables are shown configured to a 128 megabyte system. The PFA will be masked to support the CM configuration of the system being tested.

80000285 A

LMT3/P - 267

Conditions:

- C.9 Report current SVA.  
a. Advance current SVA (+200(HEX)).  
b. Set RF2 to current SVA.  
c. Report current SVA; informational.
- C.10 Test operation result.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not 0555 5400 0100 0000(HEX).\*\*
- C.11 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Advance current SVA (+200(HEX)).

If no error and current SVA is not 0--04000(HEX), return to C.3.

If no error and current set is not 2, advance set number and return to C.0.

\*\*Variables are shown configured to a 128 megabyte system. The PFA will be masked to support the CM configuration of the system being tested.

60000285 A

LMT3/P - 268

SECTION 25 - PAGE MAP ADDRESSING

Test the Page Map address bits, both external and internal to the Map RAMs.

Subsection 0 - Set 0 Page Map Addressing.

Subsection 1 - Set 1 Page Map Addressing.

Subsection 2 - Set 2 Page Map Addressing.

Method:

A forward and reverse march is sent through the 32-location Page Map RMAs. This is accomplished by first filling all Map locations with a zero pattern. Then starting at location 0 in the set being tested, read and check for zeros, followed by a write of ones. This process is repeated to each sequential location until location 32 has been tested. The reverse march is then executed by reading address 32, testing for ones and then writing zeros. The address is decremented and the procedure repeated until address zero is tested.

The writing and reading of the Page Map is accomplished using the following procedures.

The write map procedure is executed by setting up a PTD in CM and initiating a CM Write Operation. The resulting Page Table Hit will allocate and write the Page Map.

The read map procedure is executed by issuing a Test Operation using the desired SVA. The Page Map Tag is read and a Map hit results. The PFA (map) and the P0 are merged and the resulting RMA is sent back to the Register File.

An error is determined by checking the Test Operation data for the correct RMA.

Section 25, Subsection 0

This subsection tests the address logic to Page Map Set 0. It tests both the internal and external RAM addressing.

Hardware Tested:

- Shift Output to Page Map.
- MAP RAM Addressing.

Method:

Refer to section description.

60000285 A

LMT3/P - 269

# Test Patterns (hexadecimal):

## Zero Pattern Variables

| SVA            | PTD RMA | PTD                 | Page Map Address |
|----------------|---------|---------------------|------------------|
| 0000 0000 0000 | 001000  | B000 0000 0000 0000 | 0                |
| 0000 0000 0200 | 001002  | B000 0000 0040 0000 | 1                |
| 0000 0000 0400 | 001004  | B000 0000 0080 0000 | 2                |
| 0000 0000 0800 | 001008  | B000 0000 00C0 0000 | 3                |
| ↓              | ↓       | ↓                   | ↓                |
| 0000 0000 3E00 | 00103E  | B000 0000 07C0 0000 | 1F               |

## Ones Pattern Variables

| SVA            | PTD RMA | PTD                   | Page Map Address |
|----------------|---------|-----------------------|------------------|
| FFFF 7FFF C000 | 00203E  | BFFF FFFF F803 FFFF** | 0                |
| FFFF 7FFF C200 | 00203C  | BFFF FFFF F843 FFFF** | 1                |
| FFFF 7FFF C400 | 00203A  | BFFF FFFF F883 FFFF** | 2                |
| FFFF 7FFF C800 | 002038  | BFFF FFFF F8C3 FFFF** | 3                |
| ↓              | ↓       | ↓                     | ↓                |
| FFFF 7FFF FE00 | 002000  | BFFF FFFF FFC3 FFFF** | 1F               |

\*\*The PFA is shown as if testing a 128 megabyte CM configuration. This field is masked to reflect the CM configuration of the system being tested.

The CM write pattern is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = 00 for set 0.  
y--y = Current SVA.

The expected result during the forward march is 0000 0000 0000 0000(HEX).

The expected result during the reverse march is 0xFF FE00 0000 0000(HEX) where:

x = 0 for memory size of 16 MB or less.  
x = 1 for memory size of 32 MB.  
x = 3 for memory size of 64 MB.  
x = 7 for memory size of 128 MB.  
etc.

## Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 12     | Purge Map All.     |
| 15     | Test Operation.    |

60000285 A

LMT3/P - 270

## Initialization:

- Set PTA to 8; use Page Table at CM 1000(HEX).
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Set up zeros Page Table starting at CM 1000(HEX).
- Set up ones Page Table starting at CM 2000(HEX).
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX) (xxx = 0 through FFF(HEX)).
- Purge Page Map All.
  - Start processor at micrand 12, purge map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set current SVA to zero.
- Select Map Set 0.

## Conditions:

- C.0 Current write; 0s SVA.
- Set RF2 to current 0s SVA.
  - Report current 0s SVA; informational.
- C.1 Current write data.
- Set RFA to formatted write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.

Clear PTD just used.

Continue to next condition if error, SSM, or if 32 write operations complete; else advance SVA and return to C.0.

Clear current SVA, Set PTA to 10(HEX) (uses Page Table at CM 2000(HEX)).

- C.2 Current forward march SVA.
- Set RF2 to current; 0s SVA.
  - Report current SVA; informational.
- C.3 Results of test operation.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0.
- C.4 Current write; 1s SVA.
- Set RF2 to current 1s SVA.
  - Report current 1s SVA; informational.

60000285 A

LMT3/P - 271

# Conditions:

- C.5 Result of write CM.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM FFFFC0(HEX),\*\* report data; error if not as written.

Clear PTD just used.

Continue to next condition if error, SSM, or if 32 addresses tested; else advance 1s SVA and return to C.2.

Set up Zeros Page Table starting at CM 1000(HEX).

Set PTA to 8; use Page Table at CM 1000(HEX).

Set current SVA to 0000 0000 3E00(HEX).

- C.6 Current reverse march SVA.  
a. Set RF2 to current 1s SVA.  
b. Report current 1s SVA; informational.
- C.7 Results of test operation.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not 0xFF FE00 0000 0000(HEX).\*\*
- C.8 Current write; 0s SVA.  
a. Set RF2 to current 0s SVA.  
b. Report current 0s SVA; informational.
- C.9 Result of write CM.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM 0, report data; error if not as written.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 32 addresses tested; else decrement SVA and return to C.6.

\*\*The PFA is shown as if testing a 16 megabyte CM configuration. This field is masked to reflect the CM configuration of the system being tested. See page 270 for the configuration of x.

80000285 A

LMT3/P - 272

## Section 25, Subsection 1

This subsection tests the address logic to Page Map Set 1. It tests both the internal and external RAM addressing.

### Hardware Tested:

- Shift Output to Page Map.
- Page Map RAM Addressing.

### Method:

Refer to section description.

### Test Patterns:

#### Zeros Pattern Variables

| SVA            | PTD RMA | PTD                 | Page Map Address |
|----------------|---------|---------------------|------------------|
| 0000 0000 0000 | 001000  | B000 0000 0000 0000 | 0                |
| 0000 0000 0200 | 001002  | B000 0000 0040 0000 | 1                |
| 0000 0000 0400 | 001004  | B000 0000 0080 0000 | 2                |
| 0000 0000 0800 | 001008  | B000 0000 00C0 0000 | 3                |
| ↓              | ↓       | ↓                   | ↓                |
| 0000 0000 3E00 | 00103E  | B000 0000 07C0 0000 | 1F               |

#### Ones Pattern Variables

|                |        |                       |    |
|----------------|--------|-----------------------|----|
| FFFF 7FFF C000 | 00203E | BFFF FFFF F803 FFFF** | 0  |
| FFFF 7FFF C200 | 00203C | BFFF FFFF F843 FFFF** | 1  |
| FFFF 7FFF C400 | 00203A | BFFF FFFF F883 FFFF** | 2  |
| FFFF 7FFF C800 | 002038 | BFFF FFFF F8C3 FFFF** | 3  |
| ↓              | ↓      | ↓                     | ↓  |
| FFFF 7FFF FE00 | 002000 | BFFF FFFF FFC3 FFFF** | 1F |

\*\*The PFA is shown as if testing a 128 megabyte CM configuration. This field is masked to reflect the CM configuration of the system being tested.

The CM write pattern is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = 01 for set 1.  
y--y = Current SVA.

The expected result during the forward march is 0000 0000 0000 0000(HEX).  
The expected result during the reverse march is 0xFF FE00 0000 0000(HEX).\*\*\*

\*\*\*This address is shown as if testing a 128-MB CM configuration. It will be masked off to reflect the CM configuration of the system being tested.

80000285 A

LMT3/P - 273

Micrands Used:

| Number | Description                       |
|--------|-----------------------------------|
| E      | Write Word to SVA.                |
| 12     | Purge Map All 15, Test Operation. |

Initialization:

- Set PTA to 8; use Page Table at CM 1000(HEX).
- Set up zeros Page Table starting at CM 1000(HEX).
- Set up ones Page Table starting at CM 2000(HEX).
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX) (xxx = 0 through FFF(HEX)).
- Set current SVA to 0.
- Select Map Set 1.

Conditions:

- C.0 Current write 0s SVA.
- a. Set RF2 to current 0s SVA.
  - b. Report current 0s SVA; informational.
- C.1 Current write data.
- a. Set RFA to formatted write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 0, report data; error if not as written.

Clear PTD just used.

Continue to next condition if error, SSM, or if 32 write operations complete; else advance SVA and return to C.0.

Clear current SVA.

Set PTA to 10(HEX); use Page Table at CM 2000(HEX).

- C.2 Current forward march SVA.
- a. Set RF2 to current (0s) SVA.
  - b. Report current SVA; informational.
- C.3 Results of test operation.
- a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand 15, Test, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not 0.
- C.4 Current write; 1s SVA.
- a. Set RF2 to current 1s SVA.
  - b. Report current 1s SVA; informational.

80000285 A

LMT3/P - 274

Conditions:

- C.5 Result of write CM.
- a. Set RFA to current write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM FFFFC0(HEX)\*\*; report data; error if not as written.

Clear PTD just used.

Continue to next condition if error, SSM, or if 32 addresses tested; else advance ones's SVA and return to C.2.

Set up Zeros Page Table starting at CM 1000(HEX).

Set PTA to 8; use Page Table at CM 1000(HEX).

Set current SVA to 0000 0000 3E00(HEX).

- C.6 Current reverse march SVA.
- a. Set RF2 to current 1s SVA.
  - b. Report current 1s SVA; informational.
- C.7 Results of test operation.
- a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand 15, Test, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not 0xFF FE00 0000 0000(HEX).\*\*
- C.8 Current write 0s SVA.
- a. Set RF2 to current 0s SVA.
  - b. Report current 0s SVA; informational.
- C.9 Result of write CM.
- a. Set RFA to current write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 0, report data; error if not as written.
- C.10 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 32 addresses tested; else decrement SVA and return to C.6.

\*\*See page 270 for the configuration of x.

80000285 A

LMT3/P - 275

## Section 25, Subsection 2

This subsection tests the address logic to Page Map Set 2. It tests both the internal and external RAM addressing.

### Hardware Tested:

- Shift Output to Page Map.
- Page Map RAM Addressing.

### Method:

Refer to section description.

### Test Patterns:

#### Zeros Pattern Variables

| SVA            | PTD RMA | PTD                 | Page Map Address |
|----------------|---------|---------------------|------------------|
| 0000 0000 0000 | 001000  | 8000 0000 0000 0000 | 0                |
| 0000 0000 0200 | 001002  | 8000 0000 0040 0000 | 1                |
| 0000 0000 0400 | 001004  | 8000 0000 0080 0000 | 2                |
| 0000 0000 0800 | 001008  | 8000 0000 00C0 0000 | 3                |
| ↓              | ↓       | ↓                   | ↓                |
| 0000 0000 3E00 | 00103E  | 8000 0000 07C0 0000 | 1F               |

#### Ones Pattern Variables

|                |        |                       |    |
|----------------|--------|-----------------------|----|
| FFFF 7FFF C000 | 00203E | BFFF FFFF F803 FFFF** | 0  |
| FFFF 7FFF C200 | 00203C | BFFF FFFF F843 FFFF** | 1  |
| FFFF 7FFF C400 | 00203A | BFFF FFFF F883 FFFF** | 2  |
| FFFF 7FFF C600 | 002038 | BFFF FFFF F8C3 FFFF** | 3  |
| ↓              | ↓      | ↓                     | ↓  |
| FFFF 7FFF FE00 | 002000 | BFFF FFFF FFC3 FFFF** | 1F |

The CM write pattern is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = 02 for set 2.  
y--y = Current SVA.

\*\*The PFA or Address is shown as if testing a 16 megabyte CM configuration. This field is masked to reflect the CM configuration of the system being tested. See page 270 for configurations.

60000285 A

LMT3/P - 276

### Conditions:

The expected result during the forward march is 0000 0000 0000 0000(HEX).  
The expected result during the reverse march is 0xFF FE00 0000 0000(HEX).\*\*\*

### Micrands Used:

| Number | Description                       |
|--------|-----------------------------------|
| E      | Write Word to SVA.                |
| 12     | Purge Map All 15, Test Operation. |

### Initialization:

- Set PTA to 8; use Page Table at CM 1000(HEX).
- Set up zeros Page Table starting at CM 1000(HEX).
- Set up ones Page Table starting at CM 2000(HEX).
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX) (xxx = 0 through FFF(HEX)).
- Set current SVA to zero.
- Select Map Set 2.

### Conditions:

- C.0 Current write; 0s SVA.
- Set RF2 to current 0s SVA.
  - Report current 0s SVA; informational.
- C.1 Current write data.
- Set RFA to formatted write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.

Clear PTD just used.

Continue to next condition if error, SSM, or if 32 write operations complete; else advance SVA and return to C.0.

Clear current SVA.

Set PTA to 10(HEX); use Page Map at CM 2000(HEX).

- C.2 Current forward march SVA.
- Set RF2 to current; 0s SVA.
  - Report current SVA; informational.

\*\*\*The address is shown as if testing a 128 MB CM configuration. This address is marked off to reflect the CM configuration of the system being tested.

60000285 A

LMT3/P - 277

Conditions:

- C.3 Results of test operation.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not 0.
- C.4 Current write; 1s SVA.  
a. Set RF2 to current 1s SVA.  
b. Report current 1s SVA; informational.
- C.5 Result of write CM.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM FFFFC0(HEX)+), report data; error if not as written.

Clear PTD just used.

Continue to next condition if error, SSM, or if 32 addresses tested; else advance 1s SVA and return to C.2.

Set up Zeros Page Table starting at CM 1000(HEX).

Set PTA to 8; use Page Table at CM 1000(HEX).

Set current SVA to 0000 0000 3E00(HEX).

- C.6 Current reverse march SVA.  
a. Set RF2 to current 1s SVA.  
b. Report current 1s SVA; informational.
- C.7 Results of test operation.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not 0xFF FE00 0000 0000(HEX).\*\*
- C.8 Current write (0s) SVA.  
a. Set RF2 to current 0s SVA.  
b. Report current 0s SVA; informational.
- C.9 Result of write CM.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM 0, report data; error if not as written.
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

\*\*See page 270 for configurations of x.

60000285 A

LMT3/P - 278

Conditions:

- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 32 addresses tested; else decrement SVA and return to C.6.

60000285 A

LMT3/P - 279

## SECTION 26 - PURGE PAGE MAP ALL

Each set is tested individually to ensure that after a Purge Page Map All operation all locations in the Page Map have:

1. The Not Valid bit set.
2. The Tag bits all set.

Subsection 0 - Set 0 Purge Page Map All Test.

Subsection 1 - Set 1 Purge Page Map All Test.

Subsection 2 - Set 2 Purge Page Map All Test.

### Method:

The Purge Page Map All operation must be tested to guard against the possibility of false Page Map hits.

The method used writes the entire selected set with known data, purges the set, clears the entire Page Table in CM and then checks every MAP location sequentially for a hit. The Test Operation is used to check for a Page Map hit. The result from the Test Operation should show the current RMA and the Page Search Without Find (PSWF) status bit set.

A second pass through the Page Map is made using a Tag of all ones. This is used to verify that the Not Valid bit is set in all locations. Again, the result should show the PSWF status. The Purge Page Map All function code can be any one of four hexadecimal codes: C, D, E, or F. To test all four values each subsection uses a different value. To use different purge codes it is necessary to submit the purge function by way of a virtual instruction. Instruction Fetch (IF) is used to fetch the purge instruction from CM address zero, extract the code and pass it on to LM.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Set purge micrand (Micrand 28) into Control Store at address 105(HEX).

### Section 26, Subsection 0

This subsection tests Page Map Set 0 to ensure all locations are correctly purged. This includes setting the Not Valid bit and setting all bits in the Page Map Tag.

60000285 A

LMT3/P - 280

### Hardware Tested:

- Purge Control.
- Purge Clear to RAMs.

### Method:

Refer to section description.

### Test Patterns (hexadecimal):

First Pass Variables, zeros Page Map Tag.

| Page Map Address | Zeros SVAs     | PTD Address | PTD Entry           |
|------------------|----------------|-------------|---------------------|
| 0                | 0000 0000 0000 | 1000        | B000 0000 0000 0000 |
| 1                | 0000 0000 0200 | 1002        | B000 0000 0040 0000 |
| 2                | 0000 0000 0400 | 1004        | B000 0000 0080 0000 |
| 3                | 0000 0000 0600 | 1006        | B000 0000 00C0 0000 |
|                  |                |             |                     |
| v                | v              | v           | v                   |
| 1F               | 0000 0000 3E00 | 103E        | B000 0000 07C0 0000 |

Second Pass SVAs all ones Page Map Tag.

| Page Map Address | Ones SVAs      |
|------------------|----------------|
| 0                | FFFF 7FFF C000 |
| 1                | FFFF 7FFF C200 |
| 2                | FFFF 7FFF C400 |
| 3                | FFFF 7FFF C600 |
|                  |                |
| v                | v              |
| 1F               | FFFF 7FFF FE00 |

CM Write pattern: C3xx yyyy yyyy yyyy(HEX).

xx = 00 for set 0.  
y--y = Current SVA.

### Micrands Used:

| Number | Description                 |
|--------|-----------------------------|
| E      | Write Word to SVA.          |
| 10     | Set P Descriptor register.  |
| 12     | Purge Map All.              |
| 13     | Enter P register; start IF. |
| 15     | Test Operation.             |
| 28     | Virtual Purge Operation.    |

60000285 A

LMT3/P - 281



#### Initialization:

- Set up Zeros Page Table starting at CM 100016).
- Select Map set 0.
- Set P Descriptor register to Zero.
  - Set RF11 to 0--0; Segment number.
  - Set CM 0 to 0--0; P Descriptor Value.
  - Start processor at micrand 10, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1 (FE18).

#### Conditions:

- C.0 Current write SVA.  
a. Set RF2 to current 0s SVA.  
b. If Scope Mode not set, go to step c; else restore CM location 0 and the PTD from the last iteration.  
c. Report current SVA; informational.
- C.1 Current write data.  
a. Set RFA to formatted write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM 0, report data; error if not as written.

Clear PTD just used after saving it for possible use in C.0.

Continue to next condition if error, SSM, or if 32 Map locations written; else advance SVA (+200(HEX)) and return to C.0.

Clear current SVA.

- C.2 Issue Purge Map All (use function code = C).  
a. Set CM 0 to 050C0--0(HEX) after saving it for possible use in C.0.; purge instruction.  
b. Set RF10 to 0--0; P register value.  
c. Start processor at micrand 13, set P register (Start IF), wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report purge instruction; informational.
- C.3 Current read SVA; 0s pattern.  
a. Set RF2 to current 0s SVA.  
b. Report current SVA; informational.
- C.4 Result of test operation.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF6, report data; error if not 0000 xxxx 8000 0000(HEX). (xxxx = PTD address, 8008(HEX) to 81F8(HEX).)

60000285 A

LMT3/P - 282

#### Conditions:

Continue to next condition if error, SSM, or if 32 Map locations tested; else advance SVA (+200(HEX)) and return to C.3.

- C.5 Current read SVA; 1s pattern.  
a. Set RF2 to current 1s SVA.  
b. Report current SVA; informational.
- C.6 Result of test operation.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand 15, Test, wait for CPU halted.  
c. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
d. Read RF6, report data; error if not 0000 xxxx 8000 0000(HEX); xxxx = PTD address, 8008(HEX) to 81F8(HEX).
- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 32 Map locations tested; else advance ones SVA (+200(HEX)) and return to C.5.

#### Section 26, Subsection 1

Test Page Map Set 1 to ensure all locations are correctly purged. This includes setting the Not Valid bit and setting all bits in the Page Map Tag.

#### Hardware Tested:

- Purge Control.
- Purge Clear to RAMs.

#### Method:

Refer to section description.

60000285 A

LMT3/P - 283

# Test Patterns (hexadecimal):

## First Pass Variables, Zero Page Map Tag.

| Page Map Address | Zeros SVAs     | PTD Address | PTD Entry           |
|------------------|----------------|-------------|---------------------|
| 0                | 0000 0000 0000 | 1000        | B000 0000 0000 0000 |
| 1                | 0000 0000 0200 | 1002        | B000 0000 0040 0000 |
| 2                | 0000 0000 0400 | 1004        | B000 0000 0080 0000 |
| 3                | 0000 0000 0600 | 1006        | B000 0000 00C0 0000 |
|                  |                |             |                     |
| v                | v              | v           | v                   |
| 1F               | 0000 0000 3E00 | 103E        | B000 0000 07C0 0000 |

## Second Pass SVAs all Ones Page Map Tag.

| Page Map Address | Ones SVAs      |
|------------------|----------------|
| 0                | FFFF 7FFF C000 |
| 1                | FFFF 7FFF C200 |
| 2                | FFFF 7FFF C400 |
| 3                | FFFF 7FFF C600 |
|                  |                |
| v                | v              |
| 1F               | FFFF 7FFF FE00 |

CM Write pattern: C3xx yyyy yyyy yyyy(HEX).

xx = 01 for set 1.  
y--y = Current SVA.

## Micrands Used:

| Number | Description                 |
|--------|-----------------------------|
| E      | Write Word to SVA.          |
| 10     | Set P Descriptor register.  |
| 12     | Purge Map All.              |
| 13     | Enter P register; start IF. |
| 15     | Test Operation.             |
| 28     | Virtual Purge Operation.    |

## Initialization:

- Set up Zeros Page Table starting at CM 1000(HEX).
- Select Map set 1.

60000285 A

LMT3/P - 284

## Conditions:

- C.0 Current write SVA.
- Set RF2 to current 0s SVA.
  - If Scope Mode not set, go to step c; else restore CM location 0 and the PTD from the last iteration.
  - Report current SVA; informational.
- C.1 Current write data.
- Set RFA to formatted write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.

Clear PTD just used after saving it for possible use in C.0.

Continue to next condition if error, SSM, or if 32 Map locations written; else advance SVA (+200(HEX)) and return to C.0.

Clear current SVA.

- C.2 Issue Purge Map All; use function code = D.
- Set CM 0 to 050D0--0(HEX) after saving it for possible use in C.0.; purge instruction.
  - Set RF10 to 0--0; P Register value.
  - Start processor at micrand 13, set P register; start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report purge instruction; informational.
- C.3 Current read SVA; 0s pattern.
- Set RF2 to current 0s SVA.
  - Report current SVA; informational.
- C.4 Result of test operation.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF6, report data; error if not 0000 0000 8000 0000(HEX).

Continue to next condition if error, SSM, or if 32 Map locations tested; else advance SVA (+200(HEX)) and return to C.3.

- C.5 Current read SVA; 1s pattern.
- Set RF2 to current 1s SVA.
  - Report current SVA; informational.
- C.6 Result of test operation.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF6, report data; error if not 0000 0000 8000 0000(HEX).

60000285 A

LMT3/P - 285

# Conditions:

- C.7 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 32 Map locations tested; else advance ones SVA (+200(HEX)) and return to C.5.

## Section 28, Subsection 2

Test Page Map Set 2 to ensure all locations are correctly purged. This includes setting the Not Valid bit and setting all bits in the Page Map Tag.

### Hardware Tested:

- Purge Control.
- Purge Clear to RAMs.

### Method:

Refer to section description.

### Test Patterns (hexadecimal):

#### First Pass Variables, Zero Page Map Tag.

|    | Page Map |      | Zeros | SVAs | PTD     |      | PTD Entry |
|----|----------|------|-------|------|---------|------|-----------|
|    | Address  |      |       |      | Address |      |           |
| 0  | 0000     | 0000 | 0000  | 1000 | 8000    | 0000 | 0000      |
| 1  | 0000     | 0000 | 0200  | 1002 | 8000    | 0000 | 0040      |
| 2  | 0000     | 0000 | 0400  | 1004 | 8000    | 0000 | 0080      |
| 3  | 0000     | 0000 | 0600  | 1006 | 8000    | 0000 | 00C0      |
|    |          |      |       |      |         |      |           |
| v  |          | v    |       | v    |         | v    |           |
| 1F | 0000     | 0000 | 3E00  | 103E | 8000    | 0000 | 07C0      |

## Second Pass SVAs all Ones Page Map Tag.

| Page Map | Ones SVA |           |
|----------|----------|-----------|
| Address  |          |           |
| 0        | FFFF     | 7FFF C000 |
| 1        | FFFF     | 7FFF C200 |
| 2        | FFFF     | 7FFF C400 |
| 3        | FFFF     | 7FFF C600 |
|          |          |           |
| v        | v        | v         |
| 1F       | FFFF     | 7FFF FE00 |

CM Write pattern: C3xx yyyy yyyy yyyy(HEX).

xx = 02 for set 2.  
y--y = Current SVA.

### Micrands Used:

| Number | Description                 |
|--------|-----------------------------|
| E      | Write Word to SVA.          |
| 10     | Set P Descriptor register.  |
| 12     | Purge Map All.              |
| 13     | Enter P register; start IF. |
| 15     | Test Operation.             |
| 28     | Virtual Purge Operation.    |

### Initialization:

- Set up Zeros Page Table starting at CM 1000(HEX).
- Select Map set 2.

### Conditions:

- C.0 Current write SVA.
  - a. Set RF2 to current 0s SVA.
  - b. If Scope Mode not set, go to step c; else restore CM location 0 and the PTD from the last iteration.
  - c. Report current SVA; informational.
- C.1 Current Write Data.
  - a. Set RFA to formatted write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 0, report data; error if not as written.

Clear PTD just used after saving it for possible use in C.0.

Conditions:

Continue to next condition if error, SSM, or if 32 Map locations written; else advance SVA (+200(HEX)) and return to C.0.

Clear current SVA.

- C.2 Issue Purge Map All; use function code = E.
- Set CM 0 to 050E0--0(HEX) after saving it for possible use in C.0.; purge instruction.
  - Set RF10 to 0--0; P Register value.
  - Start processor at micrand 13, Set P register; start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report purge instruction; informational.
- C.3 Current Read SVA; 0s Pattern.
- Set RF2 to current 0s SVA.
  - Report current SVA; informational.
- C.4 Result of test operation.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF8, report data; error if not 0000 0000 8000 0000(HEX).

Continue to next condition if error, SSM, or if 32 Map locations tested; else advance SVA (+200(HEX)) and return to C.3.

- C.5 Current read SVA; 1s Pattern.
- Set RF2 to current 1s SVA.
  - Report current SVA; informational.
- C.6 Result of test operation.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF8, report data; error if not 0000 0000 8000 0000(HEX).
- C.7 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if 32 Map locations tested; else advance ones SVA (+200(HEX)) and return to C.5.

Reset Control Store Address 105(HEX) to original micrand.

60000285 A

LMT3/P - 288

## SECTION 27 - COMPLETE MAP PURGE TEST

Complete testing of the remaining Page Map purge functions. Test Purge Map by ASID and Purge by Page operations.

Subsection 0 - Purge by ASID, Purge by Page.

Method:

Refer to subsection descriptions.

Initialization:

- Deselect all Cache Sets.

Section 27, Subsection 0

Test Purge Page Map by ASID, Purge Map by Page.

This subsection tests the remaining Purge Map functions for correct operation.

Hardware Tested:

- Purge Control.
- Purge Write Enable.
- LM Counter to Map Address.

Method:

All three Page Map sets are tested in this subsection. Condition 0 indicates the set currently being tested.

The Purge by ASID function is tested first. This is done by initializing all locations in the Map using an alternating pattern. For example, location 0 has an ASID of 0000(HEX), location 1 an ASID of FFFF(HEX), location 2 0000(HEX), location 3 FFFF(HEX) and so on. A Purge by ASID function is issued using a SVA of FFFF 0000 0000(HEX). This should purge all odd Map locations. The Map is now scanned from location 0 through 1F(HEX), using the initial SVAs, with the Test operation. A hit should occur at every even address and a miss at every odd address.

The Purge by ASID function code can be either 9(HEX) or B(HEX). When testing sets 0 and 2, a 9(HEX) is used while set 1 has a B(HEX) used.

The Purge Map by Page function is tested by simply issuing a purge to location 0. Location 0 is now read using the Test operation. A miss is expected. The Purge by Page function code can be either 8(HEX) or A(HEX). Sets 0 and 2 use a code of 8(HEX), set 1 uses A(HEX).

60000285 A

LMT3/P - 289

# Test Patterns (hexadecimal):

## Page Map Set Initialization Variables.

|    | Page Map Address |      | Zeros SVAs |      | PTD Address |      | PTD Entry |             |
|----|------------------|------|------------|------|-------------|------|-----------|-------------|
| 0  | 0000             | 0000 | 0000       | 1000 | 8000        | 8000 | 0000      | 0000        |
| 1  | FFFF             | 7FFF | C200       | 203C | 101E0       | BFFF | FFFF      | FF80 7FFF** |
| 2  | 0000             | 0000 | 0400       | 1004 | 8020        | 8000 | 0000      | 0080 0000   |
| 3  | FFFF             | 7FFF | C800       | 2038 | 101C0       | BFFF | FFFF      | FF00 7FFF** |
| ↓  | ↓                | ↓    | ↓          | ↓    | ↓           | ↓    | ↓         | ↓           |
| 1F | FFFF             | 7FFF | FE00       | 2000 | 10000       | BFFF | FFFF      | F800 7FFF** |

\*\*PFA Set to reflect CM configuration of system under test.

Write data: C3xx yyyy yyyy yyyy(HEX).

xx = Current Set.  
yy = Current SVA.

## Micrands Used:

| Number | Description                |
|--------|----------------------------|
| E      | Write Word to SVA.         |
| 10     | Set P Descriptor register. |
| 12     | Purge All Page Maps.       |
| 13     | Set P, Start IF.           |
| 15     | Test Operation.            |
| 28     | Virtual Purge Operation.   |

## Initialization:

- Set P Descriptor register to FFFF(HEX).
- Set RF11 to 0--0; segment Number.
- Set CM 0 to 0000 FFFF 0000 0000(HEX).
- Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set Purge Micrand (micrand 28) into Control Store at address 105(HEX).
- Set PTL = 0.
- Set PSM = 7F(HEX).
- Set current Map set to 0.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX), xxx = 0 through FFF(HEX).
- Set CM 21FE(HEX) to BFFF F0--0(HEX); PTD for Purge SVA.
- Issue Purge Page Maps Micrand (12).

## Conditions:

- C.0 Set being tested.
- Set up 0s Page Table at CM 1000(HEX).
  - Set up 1s Page Table at CM 2000(HEX).
  - Select current set.
  - Report current set under test; informational.
- C.1 Zero pattern to Map even address.
- Set PTA to 8.
  - Set RF2 to current 0s SVA.
  - Set RFA to current write pattern.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.
- C.2 Ones pattern to Map odd addresses.
- Set PTA to 10(HEX).
  - Set RF2 to current 1s SVA.
  - Set RFA to current write pattern.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 1FFFC0)\*, report data; error if not as written.

Continue to next condition if error or if 32 Map locations initialized; else return to C.1.

- C.3 Purge Map, ASID = FFFF(HEX).
- Deselect all Page Maps.
  - Set CM 0 to 050x0--0(HEX); x = 9(HEX) for sets 0 and 2, B(HEX) for set 1.
  - Set RF10 to 0--0(HEX); P register value, PVA.
  - Set RF2 to FFFF 0000 0000(HEX); Purge SVA.
  - Start processor at micrand 13, Set P register, start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Select current Map Set.
  - Report 0000 FFFF 0000 0000(HEX); Purge SVA; informational.

Clear Page Table at CM 1000(HEX).

Clear Page Table at CM 2000(HEX).

Clear SVA index.

- C.4 Test Map even locations, expect hit.
- Set PTA to 8.
  - Set RF2 to current 0s SVA.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0.

Conditions:

- C.5 Test Map odd locations, expect miss.
- Set PTA to 10(HEX).
  - Set RF2 to current 1s SVA.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF6, report data; error if not 000x xxxx 8000 0000(HEX);  
xxxxx = PTD Byte Address + 8.

Continue to next condition if error or if all 32 locations tested; else return to C.4.

- C.6 Purge by Page, location 0.
- Set CM 0 to 050x0--0(HEX); x = 8(HEX) for sets 0 and 2, A (HEX) for set 1.
  - Set RF2 to 0--0; purge SVA.
  - Set RF10 to 0--0; P register Value.
  - Start processor at micrand 13, Set P register, start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report 0s; purge SVA, informational only.

- C.7 Test location 0.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 15, Test, wait for CPU halted.
  - Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - Read RF6, report data; error if not 0001 0008 8000 0000(HEX).

- C.8 CPU Status Word 1.
- Report word 1 if saved; error if reported.

- C.9 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.10 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if all three sets tested; else advance Map set and return to C.0.

Reset Control Store Address 105(HEX) to original micrand.

## SECTION 28 - PAGE MAP COMPARATOR LOGIC

The Page Map Comparator is used to sense a Page Map Hit. Portions of the SVA sent to LM are compared to portions of the Page Map. If the two variables compare the RMA is built directly from the PFA (Map) and the P0.

Subsection 0 - Set 0 Page Map Comparator.

Subsection 1 - Set 1 Page Map Comparator.

Subsection 2 - Set 2 Page Map Comparator.

Method:

The basic idea of this comparator test is to slide a single bit from the lowest bit position to the highest bit position testing for hit/miss at each position.

This is accomplished by:

- Setting the desired pattern into the Page Map using a CM Write operation to enter the Map.
- Clear the PTD in CM and issue a second write to this same SVA. A hit should occur and the data should be written.
- Clear the current comparator test bit in the SVA and issue a third write. A miss should occur and CM should not be written.
- Complement the test pattern and repeat steps 1, 2, and 3.
- Move the test bit and repeat steps 1, 2, 3 and 4.

This sequence is repeated for every bit position in the Comparator.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Deselect all Cache Sets.

### Section 28, Subsection 0

This subsection tests the 33-bit Page Map Comparator in set 0 for both hit and miss conditions.

Hardware Tested:

- Set 0 Page Map Comparator.

# Method:

Refer to section description.

Test Patterns (hexadecimal):

Sliding bit (set) variables.

| Pass Count | SVA            | PTD Address in CM | PTD Entry           |
|------------|----------------|-------------------|---------------------|
| 1          | 0000 0000 4000 | 1040              | 8000 0000 0800 0000 |
| 2          | 0000 0000 8000 | 1080              | 8000 0000 1000 0000 |
| 3          | 0000 0001 0000 | 1100              | 8000 0000 2000 0000 |
| 4          | 0000 0002 0000 | 1000              | 8000 0000 4000 0000 |
| 5          | 0000 0004 0000 | 1000              | 8000 0000 8000 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 11         | 0000 4000 0000 | 1000              | 8000 0800 0000 0000 |
| 12         | 0001 0000 0000 | 1002              | 8000 1000 0000 0000 |
| 13         | 0002 0000 0000 | 1004              | 8000 2000 0000 0000 |
| 14         | 0004 0000 0000 | 1008              | 8000 4000 0000 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 18         | 0040 0000 0000 | 1080              | 8004 0000 0000 0000 |
| 19         | 0080 0000 0000 | 1100              | 8008 0000 0000 0000 |
| 1A         | 0100 0000 0000 | 1000              | 8010 0000 0000 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 20         | 4000 0000 0000 | 1000              | 8400 0000 0000 0000 |
| 21         | 8000 0000 0000 | 1000              | 8800 0000 0000 0000 |

When testing for the miss situation the test bit is turned off in the SVA. The SVA therefore becomes all zeros.

Sliding bit (cleared) variables.

| Pass Count | SVA            | PTD Address in CM | PTD Entry           |
|------------|----------------|-------------------|---------------------|
| 1          | FFFF 7FFF 8000 | 1040              | BFFF FFFF F000 0000 |
| 2          | FFFF 7FFF 4000 | 1080              | BFFF FFFF E800 0000 |
| 3          | FFFF 7FFE C000 | 1100              | BFFF FFFF D800 0000 |
| 4          | FFFF 7FFD C000 | 1000              | BFFF FFFF B800 0000 |
| 5          | FFFF 7FFB C000 | 1000              | BFFF FFFF 7800 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 11         | FFFF 3FFF C000 | 1000              | BFFF F7FF F800 0000 |
| 12         | FFFF 7FFF C000 | 1002              | BFFF EFFF F800 0000 |
| 13         | FFFD 7FFF C000 | 1004              | BFFF DFFF F800 0000 |
| 14         | FFFB 7FFF C000 | 1008              | BFFF BFFF F800 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 18         | FFBF 7FFF C000 | 1080              | BFFB FFFF F800 0000 |
| 19         | FF7F 7FFF C000 | 1100              | BFF7 FFFF F800 0000 |
| 1A         | FEFF 7FFF C000 | 1000              | BFFE FFFF F800 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 20         | BFFF 7FFF C000 | 1000              | BBFF FFFF F800 0000 |
| 21         | 7FFF 7FFF C000 | 1000              | B7FF FFFF F800 0000 |

When testing for the miss situation the test bit is turned on in the SVA. The SVA therefore is all ones (FFFF 7FFF FE00).

Write data: C3xx yyyy yyyy yyyy.

xx = Page Map Set Number.  
yy = Current SVA.

Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |

Initialization:

- Set CM; 0 through F(HEX), to C3--C300000x; x = 0 through F(HEX).
- Select Map Set 0.
- Set pass count to 1.

Conditions:

- C.0 Pass count (1 through 21(HEX)).  
a. Increment and report pass count; informational.
- C.1 Result of first write (bit set variables).  
a. Set RF2 to current SVA.  
b. Set RFA to current write data.  
c. Set current PTD entry into CM at correct PTD address.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read CM 0, report data; error if not as written.
- C.2 Result of second write; expect hit.  
a. Clear PTD in CM.  
b. Set CM 0 to C3--C3000000(HEX).  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 0, report data; error if not as written.
- C.3 Result of third write; expect miss.  
a. Set RF2 to 0--0 (SVA).  
b. Set RFA to C300--00(HEX); write data.  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
f. Read CM 0, report data; error if not C3--C3000000(HEX).
- C.4 Result of first write; bit cleared variables.  
a. Set RF2 to current 1s SVA.  
b. Set RFA to current write data.  
c. Set current PTD entry into CM at correct PTD address.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read CM 0, report data; error if not as written.
- C.5 Result of second write; expect hit.  
a. Clear PTD in CM.  
b. Set CM 0 to C3--C3000000(HEX).  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 0, report data; error if not as written.

Conditions:

- C.6 Result of third write; expect miss.  
a. Set RF2 to FFFF 7FFF C000(HEX) (SVA).  
b. Set RFA to C300 FFFF 7FFF FE00(HEX); write data.  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
f. Read CM 0, report data; error if not C3--C3000000(HEX).
- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.  
b. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 21(HEX) passes complete; else advance current operand pointer and return to C.0.

Section 28, Subsection 1

This subsection tests the 33-bit Page Map Comparator in Set 1 for both hit and miss conditions.

Hardware Tested:

- Set 1 Page Map Comparator.

Method:

Refer to section description.



Test Patterns (hexadecimal):

Sliding bit (set) variables.

| Pass Count | SVA            | PTD Address in CM | PTD Entry           |
|------------|----------------|-------------------|---------------------|
| 1          | 0000 0000 4000 | 1040              | B000 0000 0800 0000 |
| 2          | 0000 0000 8000 | 1080              | B000 0000 1000 0000 |
| 3          | 0000 0001 0000 | 1100              | B000 0000 2000 0000 |
| 4          | 0000 0002 0000 | 1000              | B000 0000 4000 0000 |
| 5          | 0000 0004 0000 | 1000              | B000 0000 8000 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 11         | 0000 4000 0000 | 1000              | B000 0800 0000 0000 |
| 12         | 0001 0000 0000 | 1002              | B000 1000 0000 0000 |
| 13         | 0002 0000 0000 | 1004              | B000 2000 0000 0000 |
| 14         | 0004 0000 0000 | 1008              | B000 4000 0000 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 18         | 0040 0000 0000 | 1080              | B004 0000 0000 0000 |
| 19         | 0080 0000 0000 | 1100              | B008 0000 0000 0000 |
| 1A         | 0100 0000 0000 | 1000              | B010 0000 0000 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 20         | 4000 0000 0000 | 1000              | B400 0000 0000 0000 |
| 21         | 8000 0000 0000 | 1000              | B800 0000 0000 0000 |

When testing for the miss situation the test bit is turned off in the SVA. The SVA therefore becomes all zeros.

Sliding bit (cleared) variables.

| Pass Count | SVA            | PTD Address in CM | PTD Entry           |
|------------|----------------|-------------------|---------------------|
| 1          | FFFF 7FFF 8000 | 1040              | BFFF FFFF F000 0000 |
| 2          | FFFF 7FFF 4000 | 1080              | BFFF FFFF E800 0000 |
| 3          | FFFF 7FFE C000 | 1100              | BFFF FFFF D800 0000 |
| 4          | FFFF 7FFD C000 | 1000              | BFFF FFFF B800 0000 |
| 5          | FFFF 7FFB C000 | 1000              | BFFF FFFF 7800 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 11         | FFFF 3FFF C000 | 1000              | BFFF F7FF F800 0000 |
| 12         | FFFE 7FFF C000 | 1002              | BFFF EFFF F800 0000 |
| 13         | FFFD 7FFF C000 | 1004              | BFFF DFFF F800 0000 |
| 14         | FFFB 7FFF C000 | 1008              | BFFF BFFF F800 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 18         | FFBF 7FFF C000 | 1080              | BFFB FFFF F800 0000 |
| 19         | FF7F 7FFF C000 | 1100              | BFF7 FFFF F800 0000 |
| 1A         | FEFF 7FFF C000 | 1000              | BFFE FFFF F800 0000 |
|            |                |                   |                     |
| v          | v              | v                 | v                   |
| 20         | BFFF 7FFF C000 | 1000              | BBFF FFFF F800 0000 |
| 21         | 7FFF 7FFF C000 | 1000              | B7FF FFFF F800 0000 |

When testing for the miss situation the test bit is turned on in the SVA. The SVA therefore is all ones; FFFF 7FFF FE00.

Write data: C3xx yyyy yyyy yyyy.

xx = Page Map Set Number.  
yy = Current SVA.

Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |

Initialization:

- Set CM; 0 through F(HEX), to C3--C300000x; x = 0 through F(HEX).
- Select Map Set 1.
- Set current operand pointer to 0.

Conditions:

- C.0 Pass count; 1 through 21(HEX).  
a. Increment and report pass count; informational.
- C.1 Result of first write; bit set variables.  
a. Set RF2 to current SVA.  
b. Set RFA to current write data.  
c. Set current PTD entry into CM at correct PTD address.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read CM 0, report data; error if not as written.
- C.2 Result of second write; expect hit.  
a. Clear PTD in CM.  
b. Set CM 0 to C3--C3000000(HEX).  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 0, report data; error if not as written.
- C.3 Result of third write; expect miss.  
a. Set RF2 to 0--0 (SVA).  
b. Set RFA to C301--00(HEX); write data.  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
f. Read CM 0, report data; error if not C3--C3000000(HEX).
- C.4 Result of first write; bit cleared variables.  
a. Set RF2 to current 1s SVA.  
b. Set RFA to current write data.  
c. Set current PTD entry into CM at correct PTD address.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read CM 0, report data; error if not as written.
- C.5 Result of second write; expect hit.  
a. Clear PTD in CM.  
b. Set CM 0 to C3--C3000000(HEX).  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 0, report data; error if not as written.

60000285 A

LMT3/P - 300

Conditions:

- C.6 Result of third write; expect miss.  
a. Set RF2 to FFFF 7FFF C000(HEX)(SVA).  
b. Set RFA to C301 FFFF 7FFF FE00(HEX); write data.  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
f. Read CM 0, report data; error if not C3--C3000000(HEX).
- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 21(HEX) passes complete; else advance current operand pointer return to C.0.

Section 28, Subsection 2

This subsection tests the 33-bit Page Map Comparator in Set 2 for both hit and miss conditions.

Hardware Tested:

- Set 2 Page map Comparator.

Method:

Refer to section description.

Test Patterns (hexadecimal):

Sliding bit (set) variables.

| Pass Count | SVA            | PTD Address In CM | PTD Entry           |
|------------|----------------|-------------------|---------------------|
| 1          | 0000 0000 4000 | 1040              | 8000 0000 0800 0000 |
| 2          | 0000 0000 8000 | 1080              | 8000 0000 1000 0000 |
| 3          | 0000 0001 0000 | 1100              | 8000 0000 2000 0000 |
| 4          | 0000 0002 0000 | 1000              | 8000 0000 4000 0000 |

60000285 A

LMT3/P - 301

| Pass Count | SVA  |      |      | PTD Address in CM | PTD Entry |      |      |      |
|------------|------|------|------|-------------------|-----------|------|------|------|
| 5          | 0000 | 0004 | 0000 | 1000              | B000      | 0000 | 8000 | 0000 |
|            |      |      |      |                   |           |      |      |      |
| 11         | 0000 | 4000 | 0000 | 1000              | B000      | 0800 | 0000 | 0000 |
| 12         | 0001 | 0000 | 0000 | 1002              | B000      | 1000 | 0000 | 0000 |
| 13         | 0002 | 0000 | 0000 | 1004              | B000      | 2000 | 0000 | 0000 |
| 14         | 0004 | 0000 | 0000 | 1008              | B000      | 4000 | 0000 | 0000 |
|            |      |      |      |                   |           |      |      |      |
| 18         | 0040 | 0000 | 0000 | 1080              | B004      | 0000 | 0000 | 0000 |
| 19         | 0080 | 0000 | 0000 | 1100              | B008      | 0000 | 0000 | 0000 |
| 1A         | 0100 | 0000 | 0000 | 1000              | B010      | 0000 | 0000 | 0000 |
|            |      |      |      |                   |           |      |      |      |
| 20         | 4000 | 0000 | 0000 | 1000              | B400      | 0000 | 0000 | 0000 |
| 21         | 8000 | 0000 | 0000 | 1000              | B800      | 0000 | 0000 | 0000 |

When testing for the miss situation the test bit is turned off in the SVA. The SVA therefore becomes all zeros.

Sliding bit (cleared) variables.

| Pass Count | SVA  |      |      | PTD Address in CM | PTD Entry |      |      |      |
|------------|------|------|------|-------------------|-----------|------|------|------|
| 1          | FFFF | 7FFF | 8000 | 1040              | BFFF      | FFFF | F000 | 0000 |
| 2          | FFFF | 7FFF | 4000 | 1080              | BFFF      | FFFF | E800 | 0000 |
| 3          | FFFF | 7FFE | C000 | 1100              | BFFF      | FFFF | D800 | 0000 |
| 4          | FFFF | 7FFD | C000 | 1000              | BFFF      | FFFF | B800 | 0000 |
| 5          | FFFF | 7FFB | C000 | 1000              | BFFF      | FFFF | 7800 | 0000 |
|            |      |      |      |                   |           |      |      |      |
| 11         | FFFF | 3FFF | C000 | 1000              | BFFF      | F7FF | F800 | 0000 |
| 12         | FFFE | 7FFF | C000 | 1002              | BFFF      | FFFF | F800 | 0000 |
| 13         | FFFD | 7FFF | C000 | 1004              | BFFF      | FFFF | F800 | 0000 |
| 14         | FFFB | 7FFF | C000 | 1008              | BFFF      | FFFF | F800 | 0000 |
|            |      |      |      |                   |           |      |      |      |
| 18         | FFBF | 7FFF | C000 | 1080              | BFFB      | FFFF | F800 | 0000 |
| 19         | FF7F | 7FFF | C000 | 1100              | BFF7      | FFFF | F800 | 0000 |
| 1A         | FEFF | 7FFF | C000 | 1000              | BFFE      | FFFF | F800 | 0000 |
|            |      |      |      |                   |           |      |      |      |
| 20         | BFFF | 7FFF | C000 | 1000              | BBFF      | FFFF | F800 | 0000 |
| 21         | 7FFF | 7FFF | C000 | 1000              | B7FF      | FFFF | F800 | 0000 |

When testing for the miss situation the test bit is turned on in the SVA. The SVA therefore is all ones; FFFF 7FFF FE00.

Write data: C3xx yyyy yyyy yyyy.

xx = Page Map Set Number.  
yy = Current SVA.

Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |

Initialization:

- Set CM; 0 through F(HEX) to C3--C300000x; x = 0 through F(HEX).
- Select Map Set 2.
- Set current operand pointer to 0.

Conditions:

- C.0 Pass count; 1 through 21(HEX).  
a. Increment and report pass count; informational.
- C.1 Result of first write; bit set variables.  
a. Set RF2 to current SVA.  
b. Set RFA to current write data.  
c. Set current PTD entry into CM at correct PTD address.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read CM 0, report data; error if not as written.
- C.2 Result of second write; expect hit.  
a. Clear PTD in CM.  
b. Set CM 0 to C3--C3000000(HEX).  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 0, report data; error if not as written.
- C.3 Result of third write; expect miss.  
a. Set RF2 to 0--0 (SVA).  
b. Set RFA to C302--00(HEX); write data.  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).  
f. Read CM 0, report data; error if not C3--C3000000(HEX).

# Conditions:

- C.4 Result of first write; bit cleared variables.
  - a. Set RF2 to current 1s SVA.
  - b. Set RFA to current write data.
  - c. Set current PTD entry into CM at correct PTD address.
  - d. Set CM 0 to C3--C3000000(HEX).
  - e. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - f. Read CPU status, format and save if error; error if not 0.
  - g. Read CM 0, report data; error if not as written.
- C.5 Result of second write; expect hit.
  - a. Clear PTD in CM.
  - b. Set CM 0 to C3--C3000000(HEX).
  - c. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read CM 0, report data; error if not as written.
- C.6 Result of third write; expect miss.
  - a. Set RF2 to FFFF 7FFF C000(HEX)(SVA)
  - b. Set RFA to C302 FFFF 7FFF FE00(HEX); write data.
  - c. Set CM 0 to C3--C3000000(HEX).
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if Word 1 not 0 or Word 2 not 0000040--0(HEX) (PSWF).
  - f. Read CM 0, report data; error if not C3--C3000000(HEX).
- C.7 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 21(HEX) passes complete; else advance current operand pointer and return to C.0.

## SECTION 29 - PAGE MAP STATUS RAM AND ALLOCATE TEST

This section tests Page Map allocate when Map locations are allocated due to Valid, Set Enabled, or Highest LRU Code. The section also tests the LRU Update Network, and the Page Map Status RAMs for addressing and data integrity.

Subsection 0 - Page Map Allocate and LRU Update.

Subsection 1 - Page Map Status RAM Data and Address Test.

### Method:

See subsection descriptions.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Select all Page Map Sets.
- Deselect all Cache Sets.

### Section 29, Subsection 0

This subsection tests all Map allocation logic and the LRU Code formation and update network.

### Hardware Tested:

- Page Map Set Allocation.
- Page Map LRU Update.

### Method:

Allocation and LRU Update are tested by allocating and writing Map sets in a prescribed order always keeping track of what the LRU codes should be. All three sets are initialized by using unique SVAs to write to the same location in each set. After this initialization process is complete the LRU codes for the initialized location should be:

| Set | Code |
|-----|------|
| 0   | 2    |
| 1   | 1    |
| 2   | 0    |

A second series of writes is now executed to perform the actual test. This write series uses the same unique SVA, thereby controlling which set is used. When this second write series (1 through 3 writes) is complete the LRU codes should be in a known configuration. To determine if the LRU codes are correct, another write is executed to a new SVA. This causes an allocation. The set with the highest LRU code should have been allocated. All other sets are turned off and the PTD in CM is cleared, a write to the new SVA is issued and a hit is expected to occur. A miss means the wrong set was allocated.

The above method is used to test the LRU Update Network when incrementing, clearing or leaving unmodified a set LRU code.

To control the various operations necessary, a control word or indicator is used to govern the second write series. Condition 2 displays the control word. There can be up to three write operations and there is always one allocate. Condition 2 is shown as:

C.2 = FFFF FFFF FFFR wxyz(HEX)

The order and Map set numbers of the writes are given in wxy.

If any of these hexadecimal characters is F(HEX), that particular write is skipped. The z digit is the set expected to be allocated.

Test Patterns (hexadecimal):

Primary SVA Variables

| Set | SVA            | PTD<br>Address | PTD                 |
|-----|----------------|----------------|---------------------|
| 0   | 0000 0000 0000 | 1000           | 8000 0000 0000 0000 |
| 1   | 0001 0000 0000 | 1002           | 8000 1000 0000 0000 |
| 2   | 0002 0000 0000 | 1004           | 8000 2000 0000 0000 |

Alternate SVA Variables

| Set | SVA            | PTD<br>Address | PTD                 |
|-----|----------------|----------------|---------------------|
| 0   | 0004 0000 0000 | 1008           | 8000 4000 0000 0000 |
| 1   | 0005 0000 0000 | 100A           | 8000 5000 0000 0000 |
| 2   | 0006 0000 0000 | 100C           | 8000 6000 0000 0000 |

Control Words

| Pass Number | Control Word        |
|-------------|---------------------|
| 1           | FFFF FFFF FFFF 0120 |
| 2           | FFFF FFFF FFFF 1201 |
| 3           | FFFF FFFF FFFF 2012 |
| 4           | FFFF FFFF FFFF F120 |
| 5           | FFFF FFFF FFFF F201 |
| 6           | FFFF FFFF FFFF F012 |
| 7           | FFFF FFFF FFFF FF20 |
| 8           | FFFF FFFF FFFF FF01 |
| 9           | FFFF FFFF FFFF FF12 |

Write data is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = Page Map Set Number.  
y--y = Current SVA.

Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 12     | Purge Map All.     |

Initialization:

- Set CM; (0 through F(HEX) = C3--C3000x(HEX); x = 0 through F(HEX).
- Set pass count to 1.

Conditions:

- C.0 Purge Map and initialize LRU code.
- Set all Page Table Descriptors into CM.
  - Start processor at micrand 12, Purge Map All, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to Primary SVA for Set 0.
  - Set RFA to current write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0; error if not as written.
  - If error, report 0--0(HEX) (SVA) and exit condition.
  - Set RF2 to Primary SVA for Set 1.
  - Set RFA to current write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0; error if not as written.
  - If error, report 0001 0000 0000(HEX) (SVA) and exit condition.
  - Set RF2 to Primary SVA for Set 2.
  - Set RFA to current write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0; error if not as written.
  - Clear all PTDs in CM.
  - Report 0002 0000 0000(HEX)(SVA).
- C.1 Pass count; 1 through 10(HEX).
- Increment and report pass count; informational.
- C.2 Control Word.
- Report current Control Word; informational.
- C.3 Result of first write, Set W.
- If set W = F(HEX), report 0s and exit condition.
  - Set RF2 to primary SVA for set W.
  - Set RFA to formatted write data.
  - Set CM 0 to C3--C3000000(HEX).
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.
- C.4 Result of second write, Set X.
- If set X = F(HEX), report 0s and exit condition.
  - Set RF2 to primary SVA for set X.
  - Set RFA to formatted write data.
  - Set CM 0 to C3--C3000000(HEX).
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.

Conditions:

- C.5 Result of third write, Set Y.
- If set Y = F(HEX), report 0s and exit condition.
  - Set RF2 to primary SVA for set Y.
  - Set RFA to formatted write data.
  - Set CM 0 to C3--C3000000(HEX).
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.
- C.6 Allocate, result of write to Set Z.
- Set RF2 to alternate SVA for Set Z.
  - Set RFA to formatted write data.
  - Set CM 0 to C3--C3000000(HEX).
  - Set the alternate PTD for set Z into CM.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.
- C.7 Check Set Z for allocation.
- Set CM 0 to C3--C3000000(HEX).
  - Clear alternate PTD for set Z in CM.
  - Select set Z only.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not as written.
- C.8 CPU Status Word 1.
- Select all Map Sets.
  - Swap primary and alternate SVAs for set Z.
  - Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit Subsection if 10(HEX) passes complete; else pick up next Control Word and return to C.1.

## Section 29, Subsection 1

This subsection tests the 2-bit LRU Code portion of the Page Map Status RAMs. It also tests Status RAM addressing in all Page Map sets.

### Hardware Tested:

- Page Map Status RAMs.

### Method:

This subsection ensures the 2-bit LRU Code functions as expected in all 32 addresses of the Status RAMs. This is accomplished by issuing five successive write operations to one Map address. The LRU Code is therefore cycled from 0 through 2 in all three sets. This is followed by an allocate to determine if the LRU Code is left as expected. If an error should occur, either the wrong set will be allocated or a Status Parity Error (MPLRPE) will occur or both. This sequence is repeated for all 32 Page Map locations. An example for one location would be:

| LRU code values | Set<br>0 | Set<br>1 | Set<br>2 |
|-----------------|----------|----------|----------|
| After Purge     | 0        | 0        | 0        |
| After 1st Write | 0        | 1        | 1        |
| After 2nd Write | 1        | 0        | 2        |
| After 3rd Write | 2        | 1        | 0        |
| After 4th Write | 0        | 2        | 1        |
| After 5th Write | 0        | 0        | 2        |
| After Allocate  | 2        | 1        | 0        |

The Allocate should be assigned to set two as its LRU code is greatest. To verify, all other sets are deselected, the Page Table Descriptor in CM is cleared and a write to CM is issued using the same SVA used to allocate set two. A miss (PSWF) indicates the wrong set was allocated.

Addressing is tested because any wrong address will cause an incorrect LRU update causing the wrong set to be allocated.

## Test Patterns (hexadecimal):

|         | Set            | SVA  | PTD<br>Address      | PTD |
|---------|----------------|------|---------------------|-----|
| 0       | 00uu 0000 vv00 | 1000 | B00u u000 0ww0 0000 |     |
| 1       | 01uu 0000 vv00 | 1000 | B01u u000 0ww0 0000 |     |
| 2       | 02uu 0000 vv00 | 1000 | B02u u000 0ww0 0000 |     |
| 3       | 03uu 0000 vv00 | 1000 | B03u u000 0ww0 0000 |     |
| 2(Alt.) | 04uu 0000 vv00 | 1000 | B04u u000 0ww0 0000 |     |

vv = 00vv vvv0 in binary. The vs represent the position of the Page Map Address. This value is advanced once per pass.

ww = 0www ww00 in binary. The w value in the PTD is set to equal the corresponding v value.

The uu value (000u uuuu) corresponds to the v and w values and is used to maintain a Hash Code of zero.

CM write data is formatted as: C3xx yyyy yyyy yyyy.

xx = Page Map Set Number.  
yy = Current SVA.

### Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 12     | Purge Map All.     |

### Initialization:

- Purge Map All.
- Start processor at micrand 12, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition to -1(FF(HEX)).
- Set CM; 0 through F(HEX), = C3--C300000x; x = 0 through F(HEX).
- Set current SVA to 0.
- Set current PTD to B0--0(HEX).
- Set pass count to 1.

### Conditions:

- C.0 Pass count (1 through 20(HEX)).
- Select all Maps Sets.
  - Report pass count; informational.

Conditions:

- C.1 Result of first write; Set 0.  
a. Set RF2 to current set 0 SVA.  
b. Set RFA to formatted write data.  
c. Set CM; 1000(HEX) to current set 0 PTD.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Clear PTD entry in CM.  
h. Read CM 0, report data; error if not as written.
- C.2 Result of second write; Set 1.  
a. Set RF2 to current set 1 SVA.  
b. Set RFA to formatted write data.  
c. Set CM; 1002(HEX) to current set 1 PTD.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Clear PTD entry in CM.  
h. Read CM 0, report data; error if not as written.
- C.3 Result of third write; Set 2.  
a. Set RF2 to current set 2 SVA.  
b. Set RFA to formatted write data.  
c. Set CM; 1004(HEX) to current set 2 PTD.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Clear PTD entry in CM.  
h. Read CM 0, report data; error if not as written.
- C.4 Result of fourth write; Set 0.  
a. Set RF2 to current set 0 SVA.  
b. Set RFA to formatted write data.  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read CM 0, report data; error if not as written.
- C.5 Result of fifth write; Set 1.  
a. Set RF2 to current set 1 SVA.  
b. Set RFA to formatted write data.  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read CM 0, report data; error if not as written.

Conditions:

- C.6 Allocate, write to Set 2.  
a. Set RF2 to current set 2 (Alt.) SVA.  
b. Set RFA to formatted write data.  
c. Set CM; 1008(HEX) to current set 2 PTD (Alt.).  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Clear PTD entry in CM.  
h. Read CM 0, report data; error if not as written.
- C.7 Verify allocation.  
a. Select Map set 2 only.  
b. Set CM 0 to C3--C3000000(HEX).  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 0, report data; error if not as written.
- C.8 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 20(HEX) passes complete; else advance SVA (+200(HEX) and PTD (+400000(HEX) and return to C.0.



## SECTION 30 - PAGE MAP PARITY LOGIC

This section tests the parity logic associated with the Page Map. This includes SPID parity, PFA from CM parity, Map Tag Out and PFA parity from the Map. The parity error signals from LM to the PFS registers are forced to test the parity error detection logic.

Subsection 0 - Page Map (SPID/PFA) Parity Circuits.

Subsection 1 - Page Map Parity Error Signals (MPASPE).

Subsection 2 - Page Map Parity Error Signal (MPPFPE).

Method:

Refer to subsection descriptions.

Initialization:

- Set PTA to 8
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Select all Map Sets.

### Section 30, Subsection 0

This subsection tests the SPID to Page Map Parity Generator and the Map Data Out (ASD0) Parity Checker. It also tests the CM bits 42 through 63) to Map Parity Generator and the Map Data Out (PFA) Parity Checker.

Hardware Tested:

- PFA Parity Generator.
- SPID (Map Tag) Parity Gen.
- PFA Parity Checker.
- Map Tag Parity Checker.

Method:

Parity checkers tested in this subsection are tested similarly to those parity checkers previously tested. For example, a repeated pattern is used and incremented to ensure all logic gates are tested.

The PFA Parity Checker uses three separate chips to check the 22 bits sent to it. The bit breakdown is 8-8-6. Because the largest segment handles 8 bits the pattern must count from 0 through FF(HEX) or 256 counts. The output of one individual checker is passed as input (parity bit) to the next where its output is passed as input to the last checker. To create varying output signals from the first 2 chips, the counting pattern is stagger started. For example:

60000285 A

LMT3/P - 314

| Pass | PFA** Portion<br>of PFD Entry |
|------|-------------------------------|
| 1    | 00 0000                       |
| 2    | 00 0001                       |
| 3    | 00 0102                       |
| 4    | 00 0203                       |

The Map Tag Parity checker is tested in a similar manner using an incrementing repeated pattern, and stagger starting the individual fields. For example:

| Pass | SVA**          |
|------|----------------|
| 1    | 0000 0000 0000 |
| 2    | 0000 0000 4000 |
| 3    | 0000 0040 8000 |
| 4    | 0000 4080 C000 |
| 5    | 0081 00C1 0000 |
| 6    | 8101 4101 4000 |

The Parity Generators are tested by default. That is, if the checkers all pass, then the generators have also passed.

The test procedure step by step is as follows:

1. Format the test SVA. The SVA holds the test pattern for the Map Tag.
2. Format the PTD Entry. The PTD holds the test pattern for the PFA checker. It must also reflect the current SVA to ensure a Page Table Hit.
3. Calculate the CM address where the PTD must go and write the PTD to CM.
4. Use the Write Word to SVA micrand to achieve a Page Table Hit. This places the SPID and the PFA (from CM) into the Page Map.
5. Use the Write Word to SVA micrand to get a Map Hit. This step sends the data through both checkers being tested.
6. Read the word written to CM and check it.
7. Read and check the CPU Status.
8. Repeat the above steps 259 times; 259 due to the staggered start.

\*\*The PFA is first formed and then masked to reflect the CM configuration of the system under test.

60000285 A

LMT3/P - 315

# Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entry             |
|------|----------------|-----------------------|
| 1    | 0000 0000 0000 | 8000 0000 0000 0000   |
| 2    | 0000 0000 4000 | 8000 0000 0800 0001   |
| 3    | 0000 0040 8000 | 8000 0008 1000 0102   |
| 4    | 0000 4080 C000 | 8000 0810 1800 0203   |
| 5    | 0081 00C1 0000 | 8008 1018 2000 0304   |
| 6    | 8101 4101 4000 | 8810 1820 2800 0405   |
| ↓    | ↓              | ↓                     |
| 100  | FE7F 7FBF C000 | 8FE7 EFF7 F800 7DFE** |
| 101  | 7EFF 3FC0 0000 | 87EF F7F8 0000 7EFF   |
| 102  | FF7F 4000 4000 | 8FF7 F800 0800 7F00   |
| 103  | 7F80 0040 8000 | 87F8 0008 1000 0001   |
| 104  | 8000 4080 C000 | 8800 0810 1800 0102   |

\*\*Bit 32 must be ignored when generating this pattern.

Write data is formatted as: xxxx yyyy yyyy yyyy.

x-x = Represents the CM word address of the PTD Entry.  
y-y = Current SVA.

## Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |

## Initialization:

- Set all CM to C3--C3xx xxxx(HEX); where x-x is the word address.
- Clear current SVA.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 104(HEX)).  
a. Report pass count; informational.
- C.1 CM Word Address of write data.  
a. Set RF2 to current SVA.  
b. Build PTD Entry from current SVA and PFA test pattern.  
c. Calculate the CM address where the PTD Entry is to be written and write it there.  
d. Calculate the RMA of the write data, report the RMA; informational.

80000285 A

LMT3/P - 316

# Conditions:

- C.2 Expected write data.  
a. Set RFA to current write data.  
b. Report current write data.
- C.3 Actual write data, load map.  
a. Start processor at micrand E, write word to SVA, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Read CM at C.1 address, report data; error if not as reported in C.2.
- C.4 Actual write data, perform test.  
a. Clear PTD Entry in CM.  
b. Reinitialize CM at address reported in C.1.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM at address reported in C.1, report data; error if not as reported in C.2.
- C.5 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.6 CPU Status Word 2.  
a. Reinitialize CM at address reported in C.1.  
b. Report word 2 if saved; error if reported.
- C.7 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 104(HEX) passes complete; else return to C.0.

## Section 30, Subsection 1

This subsection tests the parity error signals sent from LM to the PFS registers. The Map Tag status is sent on four MPASPE lines, one per set.

## Hardware Tested:

- MPASPE Parity Lines (TAG).
- Complement Tag Parity Bit.
- PFS82, bits 40 through 43.

80000285 A

LMT3/P - 317

# Method:

The Tag or SPID parity status (MPASPE) is tested by forcing a Tag containing wrong parity into the Page Map. This Tag is then accessed to produce the parity error status which is sent to MAC. The wrong parity is forced by setting bit 11 in the processor PTM register. The parity bit is complemented at the generator. This procedure must be repeated for each set.

# Method:

Three conditions are required to test each set. Condition 0 writes the bad parity data into the page map and reports the write data. Condition 1 reports CPU Status Word One from the C.0 operation. All sets are enabled during this part of the test so the expected MPASPE status will reflect the current set and all previous sets tested. Condition 2 selects only the set being tested. A second write is performed and CPU Status Word One is reported. Now only the MPASPE status for the current set is expected. If CPU Status Word Two is ever detected as nonzero, it is saved for the C.12 report.

# Test Patterns (hexadecimal):

| Set | SVA                   | PTD Entry           |
|-----|-----------------------|---------------------|
| 0   | 00F0 0000 0000 (even) | B00F 0000 0000 0000 |
| 1   | 01E0 0000 0000 (even) | B01E 0000 0000 0000 |
| 2   | 02F0 0000 0000 (odd)  | B02F 0000 0000 0000 |

Write Data is formatted as: xxxx yyyy yyyy yyyy.

x--x = PTD Entry Address in CM.  
y--y = Current SVA.

# Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 12     | Purge Map All.     |

# Initialization:

- Purge all Page Maps.

# Conditions:

# NOTE

Disregard skipped conditions on error display.

C.0 If iteration count = 1, go to C.3.

If iteration count = 2, go to C.6.

Write Data for MPASPE Test, Set 0.

- Select all page Map Sets.
- Set RF2 to 00F0 0000 0000(HEX) (SVA).
- Set RFA to 11E0 00F0 0000 0000(HEX); write data.
- Set CM; 11E0(HEX) to B00F 0000 0000 0000(HEX); PTD entry.
- Set bit 11 in processor PTM register.
- Start processor at micrand E, write word to SVA, wait for CPU halted.
- Clear bit 11 in processor PTM register.
- Read CM 0, report data; error if not 11E0 00F0 0000 0000(HEX).

C.1 Report CPU Status Word 1.

- Read CPU Status, report Word 1; error if not 0--0800000(HEX).
- Error if Word 2 not 0. If not 0 save for C.12 report.

C.2 Execute Set 0 test, report CPU Status Word 1.

- Select Map Set 0.
- Clear PTD entry at CM; 11E0(HEX).
- Set CM 0 to C3--C3000000(HEX).
- Start processor at micrand E, write word using SVA, wait for CPU halted.
- Read and format CPU Status, report Word 1; error if not 0--0800000(HEX); MPASPE (0).

If error, repeat iteration or stop on end iteration, go to C.9.

C.3 Write data for MPASPE test, Set 1.

- Select all Page Map Sets.
- Set RF2 to 01E0 0000 0000(HEX) (SVA).
- Set RFA to 11C0 01E0 0000 0000(HEX); write data.
- Set CM 11C0(HEX) to B01E 0000 0000 0000(HEX); PTD entry.
- Set bit 11 in processor PTM register.
- Start processor at micrand E, write word to SVA, wait for CPU halted.
- Clear bit 11 in processor PTM register.
- Read CM 0, report data; error if not 11C0 01E0 0000 0000(HEX).

Conditions:

- C.4 Report CPU Status Word 1.  
a. Read CPU Status, report Word 1; error if not 0--0C00000(HEX); MPASPE (0, 1).  
b. Error if Word 2 not 0. If not 0 and first word 2 error save for C.12 report.
- C.5 Execute Set 1 test, report CPU Status Word 1.  
a. Select Map Set 1.  
b. Clear PTD Entry at CM; 11C0(HEX).  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at Micrand E, Write Word using SVA, wait for CPU halted.  
e. Read and format CPU Status, report Word 1; error if not 0--0400000(HEX); MPASPE (1).

If error, repeat iteration or stop on end iteration, go to C.9.

- C.6 Write data for MPASPE test, Set 2.  
a. Select all Page Map Sets.  
b. Set RF2 to 02F0 0000 0000(HEX); SVA.  
c. Set RFA to 11E0(HEX) 02F0 0000 0000(HEX); write data.  
d. Set CM; 11E0(HEX) to B02F 0000 0000 0000(HEX); PTD entry.  
e. Set bit 11 in processor PTM register.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Clear bit 11 in processor PTM register.  
h. Read CM 0, report data; error if not 11E0 00F0 0000 0000(HEX).
- C.7 Report CPU Status Word 1.  
a. Read CPU Status, report Word 1; error if not 0--0E00000(HEX); MPASPE (0, 1, 2).  
b. Error If Word 2 not 0. If not 0 and first Word 2 error, save for C.12 report.
- C.8 Execute Set 2 test, report CPU Status Word 1.  
a. Select Map Set 2.  
b. Clear PTD Entry at CM; 11E0(HEX).  
c. Set CM 0 to C3--C3000000(HEX).  
d. Start processor at micrand E, Write Word using SVA, wait for CPU halted.  
e. Read and format CPU Status, Report Word 1; error if not 0--0200000(HEX); MPASPE (2).

Conditions:

- C.9 CPU Status Word 1 from C.0 or C.3 or C.6 or C.9.  
a. Report Word 1 if saved; error if reported.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

If not last iteration, go to C.0.

Section 30, Subsection 2

This subsection tests the parity error signal sent from LM to PFS Register 82. The PFA status is sent on the MPPFPE line.

Hardware Tested:

- MPPFPE Parity Line (PFA).
- PFS82, bit 44.

Method:

The PFA parity status (MPPFPE) is tested by setting data containing wrong parity into CM as part of the Page Table Descriptor (PTD) entry. When the PTD is matched to the SVA and the PFA is written to the Page Map, the wrong parity (even pass counts only) is carried with it. A map parity error should be sensed. On the odd pass counts, there should be no parity errors.

Test Patterns:

| Pass | PTD                 |
|------|---------------------|
| 1    | B07F 0000 0000 0000 |
| 2    | B07F 0000 0000 0001 |
| 3    | B07F 0000 0000 0003 |
| 4    | B07F 0000 0000 0007 |
| 5    | B07F 0000 0000 000F |
| 6    | B07F 0000 0000 001F |
| 7    | B07F 0000 0000 003F |
| 8    | B07F 0000 0000 007F |
| 9    | B07F 0000 0000 00FF |
| A    | B07F 0000 0000 00FE |
| B    | B07F 0000 0000 00FC |

| Pass | PTD                 |
|------|---------------------|
| C    | B07F 0000 0000 00F8 |
| D    | B07F 0000 0000 00F0 |
| E    | B07F 0000 0000 00E0 |
| F    | B07F 0000 0000 00C0 |
| 10   | B07F 0000 0000 0080 |
| 11   | B07F 0000 0000 AAAA |
| 12   | B07F 0000 0000 0100 |
| 13   | B07F 0000 0000 0300 |
| 14   | B07F 0000 0000 0700 |
| 15   | B07F 0000 0000 0F00 |
| 16   | B07F 0000 0000 1F00 |
| 17   | B07F 0000 0000 3F00 |
| 18   | B07F 0000 0000 7F00 |
| 19   | B07F 0000 0000 FF00 |
| 1A   | B07F 0000 0000 FE00 |
| 1B   | B07F 0000 0000 FC00 |
| 1C   | B07F 0000 0000 F800 |
| 1D   | B07F 0000 0000 F000 |
| 1E   | B07F 0000 0000 E000 |
| 1F   | B07F 0000 0000 C000 |
| 20   | B07F 0000 0000 8000 |
| 21   | B07F 0000 0000 5555 |
| 22   | B07F 0000 0001 0000 |
| 23   | B07F 0000 0003 0000 |
| 24   | B07F 0000 0007 0000 |
| 25   | B07F 0000 000F 0000 |
| 26   | B07F 0000 001F 0000 |
| 27   | B07F 0000 003F 0000 |
| 28   | B07F 0000 003E 0000 |
| 29   | B07F 0000 0038 0000 |
| 30   | B07F 0000 0038 0000 |
| 31   | B07F 0000 0030 0000 |
| 32   | B07F 0000 0020 0000 |

#### Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 12     | Purge Map All.     |

#### Initialization:

- Set PTA to zero.

#### Conditions:

- C.0 Write data for MPPFPE test.
- Set RF2 to 07F0 0000 0000(HEX) (SVA).
  - Set RFA to 11E0 07F0 0000 0000(HEX); write data.
  - Disable SECEDED and Parity Checking in CM by setting bits 0 and 1 in the CM Environment Control (EC) register.
  - Enable the IOU Test Mode (TM) register by setting bit 59 in the IOU EC register.
  - Write PTD to CM 11E0(HEX) written with wrong parity on even pass counts only.
  - Disable IOU TM register.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Determine the CM Word Address (RMA) from the PTD.
  - Generate the expected data of 11E0 07F0 0000 0000(HEX) or all FFs as determined by the memory size and type.
  - Read the appropriate CM address, report data; error if not equal to expected.
- C.1 Report CPU Status Word 1.
- Read CPU Status, format and report word 1; error if not zero on odd pass counts or 0--080000(HEX) (MPPFPE) on even pass counts.
- C.2 CPU Status Word 2.
- Report word 2; error if not all clear.
  - Set CM 40) to C3--C300 0040(HEX).
- C.3 PTD entry used.
- Report the Page Table Description entry used.

Exit Subsection if all passes done; else update data and go to C.0.

Purge Map All; clear wrong parity in MAP.

Start processor at micrand 12, wait for CPU halted.

## SECTION 31 - PAGE TABLE SEARCH AND BASIC BLOCK FILL

This section tests the Page Table Search logic in the normal mode using the even/odd address requests. It also tests the Block Read Address (WBKAB) entry to the AMUX and the basic Block Fill control logic.

Subsection 0 - Page Table Search, Even/Odd Address Test.

Subsection 1 - Block Read Address (WBKAB) Test.

### Method:

Refer to subsection descriptions.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).

### Section 31, Subsection 0

This subsection tests the Page Table Search control logic to ensure the Page Table is searched using the even/odd request mechanism and not the even/even mechanism used by the Index Operation.

### Hardware Tested:

- Page Table Request Control.

### Method:

The Page Table in CM can be searched using two different methods.

1. The Index Operation disables the Request Odd logic and therefore searches the Page Table one entry at a time.
2. When not executing the Index Operation the Page Table is searched by sending two CM read requests in sequence one major cycle apart. If no hit occurs from either request, two more requests are sent. This is repeated until a hit occurs or the Continue bit is clear or the limit of 32 entries has been reached. This search method uses the even/odd request control.

To test the even/odd search, both methods are individually timed. The even/odd method should take about half as long as the Index method.

60000285 A

LMT3/P - 324

A Page Table is set up in CM consisting of 32 entries. Timing is performed by initiating a micrand sequence which will loop and execute the test micrand 16 times. The micrand loop is bracketed with micrands that read and save the CMC Free Running Counter. This records the start and end time of the test loop. The reported time (in microseconds) must be within 10 percent of the expected time or an error is reported.

All Page Map Sets are disabled.

Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description                          |
|--------|--------------------------------------|
| 12     | Purge Map All.                       |
| 17     | Write Word to SVA, Execute 16 times. |
| 18     | Index Operation, Execute 16 times.   |

### Initialization:

- Set up Page Table starting at CM 1000(HEX).
- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set P Descriptor register to Zero.
  - Set RF11 to 0--0; Segment Number.
  - Set CM 0 to 0--0; P Descriptor value.
  - Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set RF13 to F(HEX); repeat loop count.
- Disable all Map Sets.

### Conditions:

- C.0 Not used, report 0s.
- C.1 Expected index time.  
a. Report expected time for Index Operation; informational.

60000285 A

LMT3/P - 325

Conditions:

- C.2 Actual index search time.
- Set RF2 to 1F00 0000 0000(HEX) (SVA).
  - Set RF6 to DF--DF(HEX); filler.
  - Clear the CMC Free Running Counter.
  - Start processor at micrand 18, Index operation, execute 18 times, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF7, start time.
  - Read RF8, end time.
  - Determine the CPU cycle time. Report time; error if not within 10 percent of the time reported in C.1.
- C.3 Index result.
- Read RF6, report data; error if not 0000 0000 xx00 0000(HEX); where xx equals the current pass count.
- C.4 Expected even/odd search time.
- Report current expected time for normal search mode; informational.
- C.5 Actual even/odd search time.
- Set RFA to C3C31F0--0(HEX); write data.
  - Set CM 0 to C3--C3000000(HEX).
  - Clear the CMC Free Running Counter.
  - Start processor at micrand 17, write word to SVA, execute 18 times, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF7, start time.
  - Read RF8, end time.
  - Determine the CPU cycle time. Report time; error if not within 10 percent of the time reported in C.4.
- C.6 Write data from CM 0.
- Read CM 0, report data; error if not as formatted in C.5.
- C.7 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Section 31, Subsection 1

This subsection tests the Cache Address path from the Tag File (TF) through the Address Mux. This path is used to provide the Cache address during a Block Read (fill) operation. The Block Fill response logic is also tested by issuing requests from Instruction Issue (IIRQE), Instruction Fetch (IF) and Address Control A-Stream (SARQE).

Hardware Tested:

- WBKAB to Address Mux E (AMUXE).
- TF(0) Cache Address RAM.
- Block Fill Response Logic.
- Block Fill to CMC TAG.
- Write Signal (CMWCH) to Cache Data RAMs.
- Write Signal (CMWCH) to Cache Tag RAMs.

Method:

This is the first subsection to issue a read operation where a Cache miss occurs. This causes a Block Fill to be initiated, bringing four words from CM to fill one Cache block. The Block Fill is not completely tested in this subsection, only that it completes without hanging up and that the first word (word 0) was transferred from CM to Cache.

Condition 0 is used to initialize the Page Map. This eliminates any Page Table searches in conjunction with a Block Fill.

Conditions 1 through 4 execute the entire test sequence. The remaining conditions repeat Conditions 1 through 4 using different patterns.

Condition 1 issues a Block Fill request from II to fill Cache block 0 in Cache Set 0.

Condition 2 issues a Block Fill request from SA to fill Cache block 0 in Cache Set 1.

Condition 3 issues a Block Fill request from IF to fill Cache block 0 in Cache Set 2 (if available; else Set 0).

Condition 4 does nothing and is kept as a spare.

Conditions 5 through 8 repeat C.1 through C.4 using a Cache block address of FF(HEX).

Conditions 9 and 10 repeat C.1 and C.2 using a Cache block address of 55(HEX).

Conditions 11 and 12 repeat C.1 and C.2 using a Cache block address of AA HEX).

By evaluating the failing conditions, should a failure occur, a particular area of hardware can be isolated.

Test Patterns (hexadecimal):

| Conditions | SVA            | PTD Entry           | RMA    |
|------------|----------------|---------------------|--------|
| 1          | 0000 0100 0000 | F000 0020 0000 0000 | 000000 |
| 2          | 0000 0200 0000 | F000 0040 0000 0000 | 000000 |
| 3          | 0000 0300 0000 | F000 0060 0000 0000 | 000000 |
| 4          | 0000 0400 0000 | F000 0080 0000 0000 | 000000 |
| 5          | 0000 0500 1FE0 | F000 00A0 0000 0000 | 0003FC |
| 6          | 0000 0600 1FE0 | F000 00C0 0000 0000 | 0003FC |
| 7          | 0000 0700 1FE0 | F000 00E0 0000 0000 | 0003FC |
| 8          | 0000 0800 1FE0 | F000 0100 0000 0000 | 0003FC |
| 9          | 0000 0900 0AA0 | F000 0120 0000 0000 | 000154 |
| A          | 0000 0A00 0AA0 | F000 0140 0000 0000 | 000154 |
| B          | 0000 0B00 1540 | F000 0160 0000 0000 | 0002A8 |
| C          | 0000 0C00 1540 | F000 0180 0000 0000 | 0002A8 |

Micrands Used:

| Number | Description                |
|--------|----------------------------|
| E      | Write Word to SVA.         |
| D      | Read Word from SVA.        |
| F      | Purge Cache All.           |
| 12     | Purge Map All.             |
| 19     | Read Byte from SVA.        |
| 10     | Set P Descriptor register. |
| 13     | Set P register, start IF.  |

Initialization:

- Select all available Cache Sets.
- Select all Map Sets.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Set PSM to 70(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).

Conditions:

- C.0 Initialize Page Map entries.
- a. Set RFA to C3C3 C3C3 00xx(HEX); xx = current RMA; write data.
  - a. Set CM 0 to DF--DF(HEX); filler.
  - b. Set RF2 to current SVA.
  - c. Set CM; 1000(HEX) to current PTD Entry.
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read CM at current RMA; error if not C3--C3 0000xx, xx = current RMA.
  - g. Continue if error or all 12 SVAs used; else advance to next SVA and return to step a.

Report current SVA.

- C.1 II Block Fill, Block 00, Cache Set 0.
- a. Clear PTD entry at CM 1000(HEX).
  - b. Set CM 0 to C3C3 0000 C3C3 0000(HEX); read data.
  - c. Set RF2 to 0000 0100 0000(HEX) (SVA).
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Select FAKECM mode.
  - g. Set RF8 to DF--DF(HEX); filler.
  - h. Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.
  - i. Read CPU status, format and save if error; error if not 0.
  - j. Deselect FAKECM mode.
  - k. Read RF8, report data; error if not C3C3 0000 C3C3 0000(HEX).
- C.2 SA Block Fill, Block 00, Cache Set 1.
- a. Set CM 0 to C3C3 0001 C3C3 0000(HEX); read data.
  - b. Set RF2 to 0000 0200 0000(HEX) (SVA).
  - c. Set RF12 to 0--07; byte length, eight bytes.
  - d. Start processor at micrand 19, read byte from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Select FAKECM mode.
  - g. Set RF8 to DF--DF(HEX); filler.
  - h. Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.
  - i. Read CPU status, format and save if error; error if not 0.
  - j. Deselect FAKECM mode.
  - k. Read RF8, report data; error if not C3C3 0001 C3C3 0000(HEX).



Conditions:

- C.3 IF Block Fill, Block 00, Set 2 if available; else Set 0.
- Set CM 0 to C3C3 0002 C3C3 0000(HEX); read data.
  - Set RF10 to 0300 0000(HEX) (PVA).
  - Set RF2 to 0000 0300 0000(HEX)(SVA).
  - Start processor at micrand 13, set P register; start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Select FAKECM mode.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Deselect FAKECM mode.
  - Read RF8, report data; error if not C3C3 0002 C3C3 0000(HEX).
- C.4 Do nothing but report zeros.
- C.5 II Block Fill, Block FF(HEX), Cache Set 0.
- Set CM 3FC(HEX) to C3C3 0000 C3C3 03FC(HEX); read data.
  - Set RF2 to 0003 0500 1FE0(HEX) (SVA).
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Select FAKECM mode.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Deselect FAKECM mode.
  - Read RF8, report data; error if not C3C3 0000 C3C3 03FC(HEX).

Conditions:

- C.6 SA Block Fill, Block FF(HEX), Cache Set 1.
- Set CM 3FC(HEX) to C3C3 0001 C3C3 03FC(HEX); read data.
  - Set RF2 to 0003 0600 1FE0(HEX) (SVA).
  - Start processor at micrand 19, read byte from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Select FAKECM mode.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Deselect FAKECM mode.
  - Read RF8, report data; error if not C3C3 0001 C3C3 03FC(HEX).
- C.7 IF Block Fill, Block FF(HEX), Set 2 if available; else Set 0.
- Set CM 3FC(HEX) to C3C3 0002 C3C3 03FC(HEX); read data.
  - Set RF10 to 0700 1FE0(HEX) (PVA).
  - Start processor at micrand 13, set P register; Start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 0700 1FE0(HEX); SVA for Cache used.
  - Select FAKECM mode.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Deselect FAKECM mode.
  - Read RF8, report data; error if not C3C3 0002 C3C3 03FC(HEX).
- C.8 Do nothing but report zeros.

Conditions:

- C.9 II Block Fill, Block 55(HEX), Cache Set 0.  
a. Set CM 154(HEX) to C3C3 0000 C3C3 00154(HEX); read data.  
b. Set RF2 to 0001 0900 0AA0(HEX) (SVA).  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Select FAKECM mode.  
f. Set RF8 to DF--DF(HEX); filler.  
g. Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Deselect FAKECM mode.  
j. Read RF8, report data; error if not C3C3 0000 C3C3 0014(HEX).
- C.10 SA Block Fill, Block 55(HEX), Cache Set 1.  
a. Set CM 154(HEX) to C3C3 0001 C3C3 00154(HEX); read data.  
b. Set RF2 to 0001 0A00 0AA0(HEX) (SVA).  
c. Start processor at micrand 19, read byte from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Select FAKECM mode.  
f. Set RF8 to DF--DF(HEX); filler.  
g. Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Deselect FAKECM mode.  
j. Read RF8, report data; error if not C3C3 0001 C3C3 00154(HEX).
- C.11 II Block Fill, Block AA(HEX), Cache Set 0.  
a. Set CM 2A8(HEX) to C3C3 0000 C3C3 002A8(HEX); read data.  
b. Set RF2 to 0002 0B00 1540(HEX) (SVA).  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Select FAKECM mode.  
f. Set RF8 to DF--DF(HEX); filler.  
g. Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Deselect FAKECM mode.  
j. Read RF8, report data; error if not C3C3 0000 C3C3 02A8(HEX).

Conditions:

- C.12 SA Block Fill, Block AA(HEX), Cache Set 1.  
a. Set CM 2A8(HEX) to C3C3 0001 C3C3 02A8(HEX); read data.  
b. Set RF2 to 0002 0C00 1540(HEX) (SVA).  
c. Start processor at micrand 19, read byte from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Select FAKECM mode.  
f. Set RF8 to DF--DF(HEX); filler.  
g. Start processor at micrand D, read word from SVA; Cache, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Deselect FAKECM mode.  
j. Read RF8, report data; error if not C3C3 0001 C3C3 02A8(HEX).
- C.13 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.15 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

## SECTION 32 - CM TO CACHE DATA PATH AND TAG FILE TEST

This section tests all areas of Tag File 0, 1, 2 and 3, except those areas handling Address Control B-Stream logic. The B-Stream is tested by diagnostic BPT3/P.

Subsection 0 - Tag File 0, II/SA Requests.

Subsection 1 - Tag File 0, II/IF Requests.

Subsection 2 - Tag File 1, II/SA Requests.

Subsection 3 - Tag File 1, II/IF Requests.

Subsection 4 - Tag File 2, II/SA Requests.

Subsection 5 - Tag File 2, II/IF Requests.

Subsection 6 - Tag File 3, II/SA Requests.

Subsection 7 - Tag File 3, II/IF Requests.

### Method:

The Page Map and Cache are both purged and a Page Table is set up starting at CM 1000(HEX). When a read operation is requested, the Page Map must first be written, or initialized, using a Page Table Descriptor (PTD) and a portion of the SPID. The RMA is then sent to CM and the Block Fill operation starts.

This section will test all four words of the block fetched from CM and written to Cache, plus the word requested and sent to the Register File. The word number of the SVA is varied from 0 through 3 to test all bits in the Word of Interest (WOI) field. The three different requests, Instruction Issue (IIRQE), A Stream (SARQE), and Instruction Fetch (IFRQE) are used to test all (except B-Stream) WOI RAMs.

Tag File 1 is tested by forcing Tag File 0 valid using bits in the processor PTM register. The same test sequence is then repeated. Tag Files 2 and 3 are tested in a similar manner by forcing lower Tag Files valid.

The Tag File Cache Address RAMs are tested by using Block Addresses of 00, FF, 55, and AA in all Tag Files.

Pass one in each subsection is used to test the CMC to Cache data path. A unique data pattern is set into CM 0 and Conditions 1 and 2 should report the pattern.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 70(HEX).
- Set RF12 to 0--07(HEX); byte length, eight bytes.
- Select all available Cache sets.
- Select all Map sets.
- Write four PTD entries starting at CM; 1000(HEX)0.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).

### Section 32, Subsection 0

This subsection tests the Instruction Issue (II) and A-Stream (SA) areas of Tag File 0.

### Hardware Tested:

- Tag File 0.
- CMC TAG to CMC.
- CM to Cache Data Path.

### Method:

See the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, and so on. See Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request via Address Control A-Stream.

### Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address.

CM to Cache Data Path uses 0--0.

# Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word From SVA. |
| F      | Purge Cache All.    |
| 12     | Purge Map All.      |
| 19     | Read Byte from SVA. |

## Initialization:

- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to 0--0.
- Set current operand pointer to 0.
- Set pass count to 1.

## Conditions:

- C.0 Pass count (1 through 4).
- a. If repeat iteration
    - Purge all cache and page maps.
  - b. If iteration count is odd, go to C.6.
  - c. Report pass count; informational.
- C.1 IIRQE read, report data from Register File.
- a. Set RF2 to current SVA.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - 0000 0000 0000 0000(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0158(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.2 Cache data, Word 0.
- a. Select FAKECM mode.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - 0000 0000 0000 0000 - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.

60000285 A

LMT3/P - 336

# Conditions:

- C.3 Cache data, Word 1.
- a. Set RF2 to current SVA plus 8.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.4 Cache data, Word 2.
- a. Set RF2 to current SVA plus 10(HEX).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0158(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.5 Cache data, Word 3.
- a. Set RF2 to current SVA plus 18(HEX).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.

If error, repeat iteration or stop on end iteration, go to C.11.

- C.6 SARQE read, report data from Register File.
- a. Deselect FAKECM mode.
  - b. Set RF2 to current SVA.
  - c. Set RF8 to DF--DF(HEX); filler.
  - d. Start processor at micrand 19, read byte from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not
    - 0000 0000 0000 0000 - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0158(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.

60000285 A

LMT3/P - 337

# Conditions:

- C.7 Cache data, Word 0.
- Select FAKECM mode.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not
    - 0000 0000 0000 0000 - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.8 Cache data, Word 1.
- Set RF2 to current SVA plus 8.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.9 Cache data, Word 2.
- Set RF2 to current SVA plus 10(HEX).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.10 Cache data, Word 3.
- Set RF2 to current SVA plus 18(HEX).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.

80000285 A

LMT3/P - 338

# Conditions:

- C.13 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if four passes done; else advance current operand pointer and return to C.0.

## Section 32, Subsection 1

This subsection tests the Instruction Issue (II) and Instruction Fetch (IF) areas of Tag File 0.

## Hardware Tested:

- Tag File 0.
- CMC TAG to CMC.
- CM to Cache Data Path.

## Method:

See the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, and so on. See Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request from Instruction Fetch.

The II test executed in subsection 0 is repeated here to provide continuity for the IF test.

## Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address.

CM to Cache Data Pattern is FF--FF(HEX).

80000285 A

LMT3/P - 339

# Micrands Used:

| Number | Description               |
|--------|---------------------------|
| D      | Read Word From SVA.       |
| F      | Purge Cache All.          |
| 10     | Enter P Descriptor.       |
| 12     | Purge Map All.            |
| 13     | Set P register; start IF. |

## Initialization:

- Set P Descriptor to zero.
  - Set RF11 to 0--0; Segment number.
  - Set CM 0 to 0--0; P Descriptor Value.
  - Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
  - Read CPU Status, format and save if error; error if not zero. If error, set failing condition to -1(FE(HEX)).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to FF--FF(HEX).
- Set current operand pointer to 0.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).
- a. If repeat iteration,
    - Load P descriptor register.
    - Purge Cache All.
    - Purge Map All.
  - b. If iteration count is odd, go to C.6.
  - c. Increment and report pass count; informational.

## Conditions:

- C.1 IIRQE read, report data from Register File.
- a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - FFFF FFFF FFFF FFFF(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.2 Cache data, Word 0.
- a. Select FAKECM mode.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - FFFF FFFF FFFF FFFF(HEX) - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.3 Cache data, Word 1.
- a. Set RF2 to current SVA plus 8.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.4 Cache data, Word 2.
- a. Set RF2 to current SVA plus 10(HEX).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.

Conditions:

- C.5 Cache data, Word 3.
- Set RF2 to current SVA plus 18(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not  
C3C3 C3C3 C300 0003(HEX) - pass 1.  
C3C3 C3C3 C300 03FF(HEX) - pass 2.  
C3C3 C3C3 C300 0157(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.

If error, repeat iteration or stop on end iteration, go to C.11.

- C.6 IFRQE read, report data from Register File.
- Deselect FAKECM mode.
  - Set RF10 to current SVA.
  - Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand 13, set P register; start IF wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not  
A5A5 A5A5 A5A5 FFFF(HEX) - pass 1.  
A5A5 A5A5 A5A5 C3C3(HEX) - pass 2, 3, 4.

- C.7 Cache data, Word 0.
- Select FAKECM mode.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not  
FFFF FFFF FFFF FFFF(HEX) - pass 1.  
C3C3 C3C3 C300 03FC(HEX) - pass 2.  
C3C3 C3C3 C300 0154(HEX) - pass 3.  
C3C3 C3C3 C300 02A8(HEX) - pass 4.

- C.8 Cache data, Word 1.
- Set RF2 to current SVA plus 8.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not  
C3C3 C3C3 C300 0001(HEX) - pass 1.  
C3C3 C3C3 C300 03FD(HEX) - pass 2.  
C3C3 C3C3 C300 0155(HEX) - pass 3.  
C3C3 C3C3 C300 02A9(HEX) - pass 4.

60000285 A

LMT3/P - 342

Conditions:

- C.9 Cache data, Word 2.
- Set RF2 to current SVA plus 10(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not  
C3C3 C3C3 C300 0002(HEX) - pass 1.  
C3C3 C3C3 C300 03FE(HEX) - pass 2.  
C3C3 C3C3 C300 0158(HEX) - pass 3.  
C3C3 C3C3 C300 02AA(HEX) - pass 4.

- C.10 Cache data, Word 3.
- Set RF2 to current SVA plus 18(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not  
C3C3 C3C3 C300 0003(HEX) - pass 1.  
C3C3 C3C3 C300 03FF(HEX) - pass 2.  
C3C3 C3C3 C300 0157(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.

- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.

- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.13 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 4 passes done; else advance current operand pointer and return to C.0.

Section 32, Subsection 2

This subsection tests the Instruction Issue (II) and A-Stream (SA) areas of Tag File 1.

Hardware Tested:

- Tag File 1.
- CMC TAG to CMC.
- CM to Cache Data Path.

60000285 A

LMT3/P - 343

## Method:

Refer to the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, etc. See Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request via Address Control A-Stream.

## Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address.

CM to Cache Data Pattern is 55--55(HEX).

## Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word From SVA. |
| F      | Purge Cache All.    |
| 12     | Purge Map All.      |
| 19     | Read Byte from SVA. |

## Initialization:

- Set Tag File 0 Valid, set bit 40 in the processor PTM register.
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to 55--55(HEX).
- Set Pass Count to 1.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).
- a. If repeat iteration,
    - Load P descriptor register.
    - Purge Cache All.
    - Purge Map All.
  - b. If iteration count is odd, go to C.6.
  - c. Report pass count; informational.
- C.1 IIRQE read, report data from Register File.
- a. Set RF2 to current SVA.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - 5555 5555 5555 5555(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0158(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.2 Cache data, Word 0.
- a. Select FAKECM mode.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - 5555 5555 5555 5555(HEX) - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.3 Cache data, Word 1.
- a. Set RF2 to current SVA plus 8.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.



Conditions:

- C.4 Cache data, Word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not  
C3C3 C3C3 C300 0002(HEX) - pass 1.  
C3C3 C3C3 C300 03FE(HEX) - pass 2.  
C3C3 C3C3 C300 0156(HEX) - pass 3.  
C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.5 Cache data, Word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not  
C3C3 C3C3 C300 0003(HEX) - pass 1.  
C3C3 C3C3 C300 03FF(HEX) - pass 2.  
C3C3 C3C3 C300 0157(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.
- If error, repeat iteration or stop on end iteration, go to C.11.
- C.6 SARQE read, report data from Register File.  
a. Deselect FAKECM mode.  
b. Set RF2 to current SVA.  
c. Set RF6 to DF--DF(HEX); filler.  
d. Start processor at micrand 19, read byte from SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF6, report data; error if not  
5555 5555 5555 5555(HEX) - pass 1.  
C3C3 C3C3 C300 03FD(HEX) - pass 2.  
C3C3 C3C3 C300 0156(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.7 Cache data, Word 0.  
a. Select FAKECM mode.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not  
5555 5555 5555 5555(HEX) - pass 1.  
C3C3 C3C3 C300 03FC(HEX) - pass 2.  
C3C3 C3C3 C300 0154(HEX) - pass 3.  
C3C3 C3C3 C300 02A8(HEX) - pass 4.

Conditions:

- C.8 Cache data, Word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not  
C3C3 C3C3 C300 0001(HEX) - pass 1.  
C3C3 C3C3 C300 03FD(HEX) - pass 2.  
C3C3 C3C3 C300 0155(HEX) - pass 3.  
C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.9 Cache data, Word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not  
C3C3 C3C3 C300 0002(HEX) - pass 1.  
C3C3 C3C3 C300 03FE(HEX) - pass 2.  
C3C3 C3C3 C300 0156(HEX) - pass 3.  
C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.10 Cache data, Word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not  
C3C3 C3C3 C300 0003(HEX) - pass 1.  
C3C3 C3C3 C300 03FF(HEX) - pass 2.  
C3C3 C3C3 C300 0157(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.11 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 4 passes done; else advance current operand pointer and return to C.0.

Section 32, Subsection 3

This subsection tests the Instruction Issue (II) and Instruction Fetch (IF) areas of Tag File 1.

Hardware Tested:

- Tag File 1.
- CMC TAG to CMC.
- CM to Cache Data Path.

Method:

Refer to the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, etc. See Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request from Instruction Fetch.

The II test executed in subsection 2 is repeated here to provide continuity for the IF test.

Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address.

CM to Cache Data Pattern is AA--AA(HEX).

Micrands Used:

| Number | Description               |
|--------|---------------------------|
| D      | Read Word From SVA.       |
| F      | Purge Cache All.          |
| 12     | Purge Map All.            |
| 13     | Set P register; start IF. |

Initialization:

- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to AA--AA(HEX).
- Set Pass Count to 1.

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).
  - a. If repeat iteration,
    - Load P descriptor register.
    - Purge Cache All.
    - Purge Map All.
  - b. If iteration count is odd, go to C.6.
  - c. Report pass count; informational.
- C.1 IIRQE read, report data from Register File.
  - a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - AAAA AAAA AAAA AAAA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.2 Cache data, Word 0.
  - a. Select FAKECM mode.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - AAAA AAAA AAAA AAAA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.

Conditions:

- C.3 Cache data, Word 1.
- Set RF2 to current SVA plus 8.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.4 Cache data, Word 2.
- Set RF2 to current SVA plus 10(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.5 Cache data, Word 3.
- Set RF2 to current SVA plus 18(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- If error, repeat iteration or stop on end iteration, go to C.11.
- C.6 IFRQE read, report data from Register File.
- Deselect FAKECM mode.
  - Set RF10 to current SVA.
  - Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand 13, set P register (start IF), wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not
    - A5A5 A5A5 A5A5 AAAA(HEX) - Pass 1.
    - A5A5 A5A5 A5A5 C3C3(HEX) - Pass 2, 3, 4.

80000285 A

LMT3/P - 350

Conditions:

- C.7 Cache data, Word 0.
- Select FAKECM mode.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - AAAA AAAA AAAA AAAA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.8 Cache data, Word 1.
- Set RF2 to current SVA plus 8.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.9 Cache data, Word 2.
- Set RF2 to current SVA plus 10(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.10 Cache data, Word 3.
- Set RF2 to current SVA plus 18(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.

80000285 A

LMT3/P - 351

Conditions:

- C.13 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 4 passes done; else advance current operand pointer and return to C.0.

Section 32, Subsection 4

This subsection tests the Instruction Issue (II) and A-Stream (SA) areas of Tag File 2.

Hardware Tested:

- Tag File 2.
- CMC TAG to CMC.
- CM to Cache Data Path.

Method:

Refer to the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, etc. See Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request via Address Control A-Stream.

Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address.

CM to Cache Data Pattern is EA--EA(HEX).

Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| F      | Purge Cache All.    |
| 12     | Purge Map All.      |
| 19     | Read Byte from SVA. |

Initialization:

- Set Tag Files 0 and 1 Valid, set bits 40 and 41 in the processor PTM register.
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to EA--EA(HEX).
- Set Pass Count to 1.

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).
- a. If repeat iteration,
    - Load P descriptor register.
    - Purge Cache All.
    - Purge Map All.
  - b. If iteration count is odd, go to C.6.
  - c. Report pass count; informational.
- C.1 IIRQE read, report data from Register File.
- a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - EAEA EAEA EAEA EAEA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0158(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.

Conditions:

- C.2 Cache data, Word 0.  
a. Select FAKECM mode.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
EAEA EAEA EAEA EAEA(HEX) - pass 1.  
C3C3 C3C3 C300 03FC(HEX) - pass 2.  
C3C3 C3C3 C300 0154(HEX) - pass 3.  
C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.3 Cache data, Word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0001(HEX) - pass 1.  
C3C3 C3C3 C300 03FD(HEX) - pass 2.  
C3C3 C3C3 C300 0155(HEX) - pass 3.  
C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.4 Cache data, Word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0002(HEX) - pass 1.  
C3C3 C3C3 C300 03FE(HEX) - pass 2.  
C3C3 C3C3 C300 0158(HEX) - pass 3.  
C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.5 Cache data, Word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0003(HEX) - pass 1.  
C3C3 C3C3 C300 03FF(HEX) - pass 2.  
C3C3 C3C3 C300 0157(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.

If error, repeat iteration or stop on end iteration, go to C.11.

Conditions:

- C.6 SARQE read, report data from Register File.  
a. Deselect FAKECM mode.  
b. Set RF2 to current SVA.  
c. Set RF8 to DF--DF(HEX); filler.  
d. Start processor at micrand 19, read byte from SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF8, report data; error if not  
EAEA EAEA EAEA EAEA(HEX) - pass 1.  
C3C3 C3C3 C300 03FD(HEX) - pass 2.  
C3C3 C3C3 C300 0158(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.7 Cache data, Word 0.  
a. Select FAKECM mode.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
EAEA EAEA EAEA EAEA(HEX) - pass 1.  
C3C3 C3C3 C300 03FC(HEX) - pass 2.  
C3C3 C3C3 C300 0154(HEX) - pass 3.  
C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.8 Cache data, Word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0001(HEX) - pass 1.  
C3C3 C3C3 C300 03FD(HEX) - pass 2.  
C3C3 C3C3 C300 0155(HEX) - pass 3.  
C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.9 Cache data, Word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0002(HEX) - pass 1.  
C3C3 C3C3 C300 03FE(HEX) - pass 2.  
C3C3 C3C3 C300 0158(HEX) - pass 3.  
C3C3 C3C3 C300 02AA(HEX) - pass 4.

Conditions:

- C.10 Cache data, Word 3.  
 a. Set RF2 to current SVA plus 18(HEX).  
 b. Set RF6 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF6, report data; error if not  
     C3C3 C3C3 C300 0003(HEX) - pass 1.  
     C3C3 C3C3 C300 03FF(HEX) - pass 2.  
     C3C3 C3C3 C300 0157(HEX) - pass 3.  
     C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.11 CPU Status Word 1.  
 a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
 a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.  
 a. Report PFS87 bits 19-23 in bits 59-63; information only.  
 b. Report last PTL written in bits 34-47; information only.  
 c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 4 passes done; else advance current operand pointer and return to C.0.

#### Section 32, Subsection 5

This subsection tests the Instruction Issue (II) and Instruction Fetch (IF) areas of Tag File 2.

Hardware Tested:

- Tag File 2.
- CMC TAG to CMC.
- CM to Cache Data Path.

Method:

Refer to the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, etc. See Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request from Instruction Fetch.

The II test executed in subsection 4 is repeated here to provide continuity for the IF test.

Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address.

CM to Cache Data Pattern is AE--AE(HEX).

Micrands Used:

| Number | Description               |
|--------|---------------------------|
| D      | Read Word From SVA.       |
| F      | Purge Cache All.          |
| 12     | Purge Map All.            |
| 13     | Set P register; start IF. |

Initialization:

- Purge Cache All.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map All.
- Start processor at micrand 12, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to AE--AE(HEX).
- Set Pass Count to 1.

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).  
 a. If repeat iteration,  
   - Load P descriptor register.  
   - Purge Cache All.  
   - Purge Map All.  
 b. If iteration count is odd, go to C.8.  
 c. Report pass count; informational.
- C.1 IIRQE read, report data from Register File.  
 a. Set RF2 to current SVA.  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not  
   AEAE AEAE AEAE AEAE(HEX) - pass 1.  
   C3C3 C3C3 C300 03FD(HEX) - pass 2.  
   C3C3 C3C3 C300 0156(HEX) - pass 3.  
   C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.2 Cache data, Word 0.  
 a. Select FAKECM mode.  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not  
   AEAE AEAE AEAE AEAE(HEX) - pass 1.  
   C3C3 C3C3 C300 03FC(HEX) - pass 2.  
   C3C3 C3C3 C300 0154(HEX) - pass 3.  
   C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.3 Cache data, Word 1.  
 a. Set RF2 to current SVA plus 8.  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not  
   C3C3 C3C3 C300 0001(HEX) - pass 1.  
   C3C3 C3C3 C300 03FD(HEX) - pass 2.  
   C3C3 C3C3 C300 0155(HEX) - pass 3.  
   C3C3 C3C3 C300 02A9(HEX) - pass 4.

Conditions:

- C.4 Cache data, Word 2.  
 a. Set RF2 to current SVA plus 10(HEX).  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not  
   C3C3 C3C3 C300 0002(HEX) - pass 1.  
   C3C3 C3C3 C300 03FE(HEX) - pass 2.  
   C3C3 C3C3 C300 0156(HEX) - pass 3.  
   C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.5 Cache data, Word 3.  
 a. Set RF2 to current SVA plus 18(HEX).  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not  
   C3C3 C3C3 C300 0003(HEX) - pass 1.  
   C3C3 C3C3 C300 03FF(HEX) - pass 2.  
   C3C3 C3C3 C300 0157(HEX) - pass 3.  
   C3C3 C3C3 C300 02AB(HEX) - pass 4.

If error, repeat iteration or stop on end iteration, go to C.11.

- C.6 IFRQE read, report data from Register File.  
 a. Deselect FAKECM mode.  
 b. Set RF10 to current SVA.  
 c. Set RF14 to DF--DF(HEX); filler.  
 d. Start processor at micrand 13, set P register; start IF, wait for CP halted.  
 e. Read CPU status, format and save if error; error if not 0.  
 f. Read RF14, report data; error if not  
   A5A5 A5A5 A5A5 AEAE(HEX) - Pass 1.  
   A5A5 A5A5 A5A5 C3C3(HEX) - Pass 2, 3, 4.
- C.7 Cache data, Word 0.  
 a. Select FAKECM mode.  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not  
   AEAE AEAE AEAE AEAE(HEX) - pass 1.  
   C3C3 C3C3 C300 03FC(HEX) - pass 2.  
   C3C3 C3C3 C300 0154(HEX) - pass 3.  
   C3C3 C3C3 C300 02A8(HEX) - pass 4.

# Conditions:

- C.8 Cache data, Word 1.
- Set RF2 to current SVA plus 8.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.9 Cache data, Word 2.
- Set RF2 to current SVA plus 10(HEX).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.10 Cache data, Word 3.
- Set RF2 to current SVA plus 18(HEX).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 4 passes done; else advance current operand pointer and return to C.0.

## Section 32, Subsection 6

This subsection tests the Instruction Issue (II) and A-Stream (SA) areas of Tag File 3.

### Hardware Tested:

- Tag File 3.
- CMC TAG to CMC.
- CM to Cache Data Path.

### Method:

Refer to the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, and so on. Refer to Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request via Address Control A-Stream.

### Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address.

CM to Cache Data Pattern is BA--BA(HEX).

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word From SVA. |
| F      | Purge Cache All.    |
| 12     | Purge Map All.      |
| 19     | Read Byte from SVA. |



# Initialization:

- Set Tag Files 0, 1 and 2 Valid, set bits 40, 41 and 42 in the processor PTM register.
- Purge Cache A11.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map A11.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to BA--BA(HEX).
- Set Pass Count to 1.

# Conditions:

## NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).
- a. If repeat iteration,
    - Load P descriptor register.
    - Purge Cache A11.
    - Purge Map A11.
  - b. If iteration count is odd, go to C.8.
  - c. Report pass count; informational.
- C.1 IIRQE read, report data from Register File.
- a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - BABA BABA BABA BABA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.2 Cache data, Word 0.
- a. Select FAKECM mode.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - BABA BABA BABA BABA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.

60000285 A

LMT3/P - 362

# Conditions:

- C.3 Cache data, Word 1.
- a. Set RF2 to current SVA plus 8.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.4 Cache data, Word 2.
- a. Set RF2 to current SVA plus 10(HEX).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.5 Cache data, Word 3.
- a. Set RF2 to current SVA plus 18(HEX).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- If error, repeat iteration or stop on end iteration, go to C.11.
- C.6 SARQE Read, Report Data from Register File.
- a. Deselect FAKECM mode.
  - b. Set RF2 to current SVA.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Start processor at micrand 19, read byte from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF6, report data; error if not
    - BABA BABA BABA BABA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.

60000285 A

LMT3/P - 363

# Conditions:

- C.7 Cache data, Word 0.
- Select FAKECM mode.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - BABA BABA BABA BABA(HEX) - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.8 Cache data, Word 1.
- Set RF2 to current SVA plus 8.
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.9 Cache data, Word 2.
- Set RF2 to current SVA plus 10(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.10 Cache data, Word 3.
- Set RF2 to current SVA plus 18(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.

60000285 A

LMT3/P - 364

# Conditions:

- C.13 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 4 passes done; else advance current operand pointer and return to C.0.

## Section 32, Subsection 7

This subsection tests the Instruction Issue (II) and Instruction Fetch (IF) areas of Tag File 3.

## Hardware Tested:

- Tag File 3.
- CMC TAG to CMC.
- CM to Cache Data Path.

## Method:

Refer to the section description for the general method.

All conditions must be executed four times. The first pass uses a Cache Block Address of zero and a starting word number of zero. The second pass uses a Block Address of FF(HEX) and a starting word number of one, etc. Refer to Test Patterns for additional detail.

Condition 1 initiates the Block Read and reports the word returned to the Register File. Conditions 2 through 5 use FAKECM mode to read the four words transferred from CM to Cache. Conditions 6 through 10 report the same respective indications as conditions 1 through 5. The one difference is condition 6 issues its request from Instruction Fetch.

The II test executed in subsection 6 is repeated here to provide continuity for the IF test.

## Test Patterns (hexadecimal):

| Pass | SVA            | PTD Entries         |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 |
| 2    | 0000 0000 1FE8 | F000 0000 01C0 0000 |
| 3    | 0000 0000 0AB0 | F000 0000 0080 0000 |
| 4    | 0000 0000 1558 | F000 0000 0140 0000 |

CM data is initialized such that each word contains its own word address. CM to Cache Data Pattern is AB--AB(HEX).

60000285 A

LMT3/P - 365

# Micrands Used:

| Number | Description               |
|--------|---------------------------|
| D      | Read Word From SVA.       |
| F      | Purge Cache A11.          |
| 12     | Purge Map A11.            |
| 13     | Set P register; start IF. |

## Initialization:

- Purge Cache A11.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map A11.
- Start processor at micrand 12, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 to AB--AB(HEX).
- Set current operand pointer to 0.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).
- a. If repeat iteration,
    - Load P descriptor register.
    - Purge Cache A11.
    - Purge Map A11.
  - b. If iteration count is odd, go to C.8.
  - c. Increment and report pass count; informational.
- C.1 IIRQE read, report data from Register File.
- a. Set RF2 to current SVA.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - ABAB ABAB ABAB ABAB(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.

## Conditions:

- C.2 Cache data, Word 0.
- a. Select FAKECM mode.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - ABAB ABAB ABAB ABAB(HEX) - pass 1.
    - C3C3 C3C3 C300 03FC(HEX) - pass 2.
    - C3C3 C3C3 C300 0154(HEX) - pass 3.
    - C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.3 Cache data, Word 1.
- a. Set RF2 to current SVA plus 8.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - C3C3 C3C3 C300 0001(HEX) - pass 1.
    - C3C3 C3C3 C300 03FD(HEX) - pass 2.
    - C3C3 C3C3 C300 0155(HEX) - pass 3.
    - C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.4 Cache data, Word 2.
- a. Set RF2 to current SVA plus 10(HEX).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - C3C3 C3C3 C300 0002(HEX) - pass 1.
    - C3C3 C3C3 C300 03FE(HEX) - pass 2.
    - C3C3 C3C3 C300 0156(HEX) - pass 3.
    - C3C3 C3C3 C300 02AA(HEX) - pass 4.
- C.5 Cache data, Word 3.
- a. Set RF2 to current SVA plus 18(HEX).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not
    - C3C3 C3C3 C300 0003(HEX) - pass 1.
    - C3C3 C3C3 C300 03FF(HEX) - pass 2.
    - C3C3 C3C3 C300 0157(HEX) - pass 3.
    - C3C3 C3C3 C300 02AB(HEX) - pass 4.

If error, repeat iteration or stop on end iteration, go to C.11.

Conditions:

- C.6 IFRQE read, report data from Register File.  
a. Deselect FAKECM mode.  
b. Set RF10 to current SVA.  
c. Set RF14 to DF--DF(HEX); filler.  
d. Start processor at micrand 13, set P register; start IF, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF14, report data; error if not.  
A5A5 A5A5 A5A5 ABAB - Pass 1.  
A5A5 A5A5 A5A5 C3C3 - Pass 2, 3, 4.
- C.7 Cache data, Word 0.  
a. Select FAKECM mode.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
ABAB ABAB ABAB ABAB(HEX) - pass 1.  
C3C3 C3C3 C300 03FC(HEX) - pass 2.  
C3C3 C3C3 C300 0154(HEX) - pass 3.  
C3C3 C3C3 C300 02A8(HEX) - pass 4.
- C.8 Cache data, Word 1.  
a. Set RF2 to current SVA plus 8.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0001(HEX) - pass 1.  
C3C3 C3C3 C300 03FD(HEX) - pass 2.  
C3C3 C3C3 C300 0155(HEX) - pass 3.  
C3C3 C3C3 C300 02A9(HEX) - pass 4.
- C.9 Cache data, Word 2.  
a. Set RF2 to current SVA plus 10(HEX).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0002(HEX) - pass 1.  
C3C3 C3C3 C300 03FE(HEX) - pass 2.  
C3C3 C3C3 C300 0156(HEX) - pass 3.  
C3C3 C3C3 C300 02AA(HEX) - pass 4.

Conditions:

- C.10 Cache data, Word 3.  
a. Set RF2 to current SVA plus 18(HEX).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not  
C3C3 C3C3 C300 0003(HEX) - pass 1.  
C3C3 C3C3 C300 03FF(HEX) - pass 2.  
C3C3 C3C3 C300 0157(HEX) - pass 3.  
C3C3 C3C3 C300 02AB(HEX) - pass 4.
- C.11 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 4 passes done; else advance current operand pointer and return to C.0.

Clear bits 40, 41 and 42 in the processor PTM register.

## SECTION 33 - CACHE HIT VALID LOGIC

This section tests the Cache Hit Valid logic to ensure each one of the four Word Address Valid bits is detected.

Subsection 0 - Cache Hit Valid, Sets 0 and 1.

Subsection 1 - Cache Hit Valid, Sets 2 and 3; if available.

### Method:

Cache Hit Valid occurs after a Cache hit during a read operation (Block Fill). This happens when the bit in the Word Address Valid RAM which represents the requested word is set. If the bit is sensed as clear due to a failure, the Hit Not Valid signal is invoked and the read micrand will hang up waiting for a response.

To test the Hit Valid bits the following procedure is used.

1. Issue a read (Cache miss) to block fill Cache block zero.
2. Change the data in CM where the Cache block was fetched from.
3. Issue a read to each word of the block read in step 1. A Cache Hit Valid should occur and the word requested from Cache should be sent to the Register File.

Requests from Instruction Issue (II), Instruction Fetch (IF), and Address Control A Stream (SA) are used to test the response logic for Hit Valid.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Select all map sets.

### Section 33, Subsection 0

This subsection tests Cache sets 0 and 1 for the Hit Valid situation using each of the four words in a Cache block.

### Hardware Tested:

- Cache Hit Valid Logic.
- Hit Valid Response Signals.

60000285 A

LMT3/P - 370

### Method:

One pass is made through the 12 conditions. Conditions 0 through 4 test set 0, conditions 5 through 9 test set 1.

Refer to section description for additional detail.

### Test Patterns:

Refer to conditions.

### Micrands Used:

| Number | Description               |
|--------|---------------------------|
| D      | Read Word from SVA.       |
| 19     | Read Bytes from SVA.      |
| 10     | Set P Descriptor.         |
| 13     | Set P register, start IF. |
| F      | Purge Cache All.          |
| 12     | Purge Map All.            |

### Initialization:

- Select Cache Sets 0 and 1.
- Set CM 1000(HEX) to F000 0000 0000 0000(HEX).
- Set CM 1001(HEX) to B000 0080 0000 0000(HEX).

### Conditions:

- C.0 Block read data from Cache Set 0, report data read.
- a. Purge Cache All.
    - Start processor at micrand F, wait for CPU halted.
    - Read CPU status, format and save if error; error if not zero. If error set failing condition number to -1(FE(HEX)).
  - b. Purge Map All.
    - Start processor at micrand 12, wait for CPU halted.
    - Read CPU status, format and save if error; error if not zero. If error set failing condition number to -1(FE(HEX)).
  - c. Set P Descriptor Register to 0--0.
    - Set RF11 to 0--0; Segment Number.
    - Set CM 0 to 0--0; P Descriptor Value.
    - Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
    - Read CPU status, format and save if error; error if not zero. If error set failing condition number to -1(FE(HEX)).
  - d. Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
  - e. Set RF2 to 0000 0000 0000(HEX) (SVA).

60000285 A

LMT3/P - 371

Conditions (C.0 continued):

- f. Set RF6 to DF--DF(HEX); filler.
  - g. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
  - j. Read RF6, report data; error if not C3--C3000000(HEX).
- C.1 Word 0 valid, test I1 response.
- a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not C3--C3000000(HEX).
- C.2 Word 1 valid, test SA response.
- a. Set RF2 to 0000 0000 0008(HEX); SVA, word 1.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Set RF12 to 0--07(HEX); byte length.
  - d. Start processor at micrand 19, read bytes from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF6, report data; error if not C3--C30000001(HEX).
- C.3 Word 2 valid, test I1 response.
- a. Set RF2 to 0000 0000 0010(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not C3--C30000002(HEX).
- C.4 Word 3 valid, test IF response.
- a. Set RF10 to 0--018(HEX) (PVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 13, transfer RF10 to the P register; start IF, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF14, report data; error if not A5A5 A5A5 A5A5 C3C3(HEX).
- Set CM; 0 through F(HEX) to C3--C300000x(HEX); x = 0 through F(HEX).
- C.5 Block read data from Cache Set 1, report data read.
- a. Set RF2 to 0000 0400 0000(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
  - f. Read RF6, report data; error if not C3--C3000000(HEX).

Conditions:

- C.6 Word 0 valid, test I1 response.
- a. Set RF6 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF6, report data; error if not C3--C3000000(HEX).
- C.7 Word 1 valid, test SA response.
- a. Set RF2 to 0000 0400 0008(HEX); SVA, word 1.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 19, read bytes from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not C3--C30000001(HEX).
- C.8 Word 2 valid, test I1 response.
- a. Set RF2 to 0000 0400 0010 (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not C3--C30000002(HEX).
- C.9 Word 3 valid, test IF response.
- a. Set RF10 to 0400 0018(HEX) (PVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand 13, transfer RF10 to the P register; start IF, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not zero.
  - e. Read RF14, report data; error if not A5A5 A5A5 A5A5 C3C3(HEX).
- C.10 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Section 33, Subsection 1

This subsection tests Cache sets 2 and 3 for the Hit Valid situation using each of the four words in a Cache block. If sets 2 and 3 are not configured, exit subsection.

#### Hardware Tested:

- Cache Hit Valid Logic.
- Hit Valid Response Signals.

#### Method:

One pass is made through the 12 conditions. Conditions 0 through 4 test set 2, conditions 5 through 9 test set 3.

Refer to section description for additional detail.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description               |
|--------|---------------------------|
| D      | Read Word from SVA.       |
| 19     | Read Bytes from SVA.      |
| 13     | Set P register, start IF. |
| F      | Purge Cache All.          |
| 12     | Purge Map All.            |

#### Initialization:

- Select Cache Sets 2 and 3 if available; else exit subsection.
- Set CM 1000(HEX) to F000 0000 0000 0000(HEX).
- Set CM 1001(HEX) to B000 0080 0000 0000(HEX).

#### Conditions:

#### NOTE

Disregard skipped conditions on error display.

#### Conditions:

- C.0 Block read data from Cache Set 2, report data read.
- Purge Cache All.
    - Start processor at micrand F, wait for CPU halted.
    - Read CPU status, format and save if error; error if not zero. If error set failing condition number to -1(FE(HEX)).
  - Purge Map All.
    - Start processor at micrand 12, wait for CPU halted.
    - Read CPU status, format and save if error; error if not zero. If error set failing condition number to -1(FE(HEX)).
  - Set CM 0 through F(HEX), to C3--C300000x(HEX); x = 0 through F (HEX).
  - If iteration count is nonzero, go to C.5.
  - Set RF2 to 0000 0000 0000(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
  - Read RF8, report data; error if not C3--C30000(HEX).
- C.1 Word 0 valid, test I1 response.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C3--C30000(HEX).
- C.2 Word 1 valid, test SA response.
- Set RF2 to 0000 0000 0008(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 19, read bytes from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C3--C30001(HEX).
- C.3 Word 2 valid, test I1 response.
- Set RF2 to 0000 0000 0010(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C3--C30002(HEX).
- C.4 Word 3 valid, test IF response.
- Set RF10 to 0--018(HEX) (PVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand 13, transfer RF10 to the P register; start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not A5A5 A5A5 A5A5 C3C3(HEX).

Conditions:

If error, repeat iteration or stop on end iteration, go to C.10.

Set CM 0 through F(HEX) to C3--C300000x(HEX); x = 0 through F(HEX).

- C.5 Block read data from Cache Set 3, report data read.  
a. Set RF2 to 0000 0400 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Set CM; 0 through 3 to Fx--Fx(HEX), where x = 0 through 3.  
f. Read RF8, report data; error if not C3--C30000(HEX).
- C.6 Word 0 valid, test I1 response.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not C3--C30000(HEX).
- C.7 Word 1 valid, test SA response.  
a. Set RF2 to 0000 0400 0008(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand 19, read bytes from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C30001(HEX).
- C.8 Word 2 valid, test I1 response.  
a. Set RF2 to 0000 0400 0010 (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C30002(HEX).
- C.9 Word 3 valid, test IF response.  
a. Set RF10 to 0400 0018(HEX) (PVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand 13, transfer RF10 to the P register; start IF, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF14, report data; error if not A5A5 A5A5 A5A5 C3C3(HEX).
- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

Conditions:

C.12 First Failure Capture Information.

- a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.



## SECTION 34 - CACHE HIT NOT VALID AND ASSOCIATED RAMS

This section tests the Hit Not Valid logic. Hit Not Valid occurs when two reads to different words of the same Cache block are executed in close proximity. The first read initiates a block fill operation, the second read results in a hit. However the word it requested is not yet in Cache. The Tag File WOI field is updated to indicate the word has been requested and the requestor is waiting for the word to be received from CM.

Two fields in the 256-location Cache Tag, the Word Address Valid RAM (4 bits) and the Tag File Address RAM (2 bits), are tested for data integrity and correct addressing.

Subsection 0 - Set 0 Cache Hit Not Valid and RAM Test.

Subsection 1 - Set 1 Cache Hit Not Valid and RAM Test.

Subsection 2 - Set 2 Cache Hit Not Valid and RAM Test.

Subsection 3 - Set 3 Cache Hit Not Valid and RAM Test.

### Method:

The Hit Not Valid logic is tested by starting a Cache Block Fill operation followed immediately by a second read to the same Cache block at a different word. For example, the first read requests word zero. This starts a block fill and words zero, one, two, and three are fetched from CM in that order. A second read requests word three of the same block. Since word three has not yet arrived the Word Valid bit for word three is still clear. This results in a Cache Hit, but the word requested is Not Valid. The operation waits for word three and sends it on to the Register File when it is received from CM. If a failure occurs and the Word Address Valid bit does not clear, the old data in Cache is returned and the error is sensed.

A double micrand sequence (two reads) is required to test one word in one block. One pass through the test conditions tests all four words in one Cache block. Therefore, 256 passes are required to test all locations in the Word Address Valid RAM.

The Tag File Address RAM is tested by using a different Tag File for each word test. This sets the current location in the Tag File Address RAM to all possible bit combinations (0 through 3). Therefore, after 256 passes all locations have been tested.

Condition 1 initializes the current Cache Block using FAKECM mode. The Cache Tag is allocated using a unique SVA and the four words in the current block are written.

## NOTE

The Tag File Address Parity checker and CABRP Parity Generator are also tested during this section. This results because all possible Cache Block addresses are passed through the checker and generator.

All Cache is purged before each test to allow valid Condition looping.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 0.
- Select all Map Sets.

### Section 34, Subsection 0

This subsection tests the Hit Not Valid logic, the Cache Tag Word Address Valid RAM, and the Cache Tag File Address RAM in Cache Set 0.

### Hardware Tested:

- Word Address Valid RAMs.
- Tag File Address RAMs.
- Hit Not Valid Logic.
- Tag File Address (TFADPE) Parity Checker.
- Cache Block Address (CABRP) Parity Generator.

### Method:

Conditions 1 and 2 are used to initialize the current Cache Block; that is, the block to be tested. The data is set to known constants and the Tag set so a miss is ensured.

Two conditions are required to test one word. Therefore conditions 3 and 4 test word 0, conditions 5 and 6 test word 1, etc.

Conditions 3, 5, 7, and 9 perform the actual test, the double read, then report the result of the first read.

Conditions 4, 6, 8, and 10 report the result of the second or Hit Not Valid read.

This subsection uses two II Requests to test the IIWOI gates when the Tag File is updated.

For additional details see the section description.

Test Patterns (hexadecimal):

| Pass | Cache Block | Word Tested | SVA of First Read | SVA of Second Read |
|------|-------------|-------------|-------------------|--------------------|
| 1    | 0           | 0           | 0000 0000 0008    | 0000 0000 0000     |
| 1    | 0           | 1           | 0000 0100 0010    | 0000 0100 0008     |
| 1    | 0           | 2           | 0000 0000 0018    | 0000 0000 0010     |
| 1    | 0           | 3           | 0000 0100 0000    | 0000 0100 0018     |
| 2    | 1           | 0           | 0000 0000 0028    | 0000 0000 0020     |
| 2    | 1           | 1           | 0000 0100 0030    | 0000 0100 0028     |
| 2    | 1           | 2           | 0000 0000 0038    | 0000 0000 0030     |
| 2    | 1           | 3           | 0000 0100 0020    | 0000 0100 0038     |
| 3    | 2           | 0           | 0000 0000 0048    | 0000 0000 0040     |
| 3    | 2           | 1           | 0000 0100 0050    | 0000 0100 0048     |
| 3    | 2           | 2           | 0000 0000 0058    | 0000 0000 0050     |
| 3    | 2           | 3           | 0000 0100 0040    | 0000 0100 0058     |
|      |             |             |                   |                    |
| v    | v           | v           | v                 | v                  |
| 100  | FF          | 0           | 0000 0000 1FE8    | 0000 0000 1FE0     |
| 100  | FF          | 1           | 0000 0100 1FF0    | 0000 0100 1FE8     |
| 100  | FF          | 2           | 0000 0000 1FF8    | 0000 0000 1FF0     |
| 100  | FF          | 3           | 0000 0100 1FE0    | 0000 0100 1FF8     |

| Pass | PTD Entry (CM(1000)) | RMA of First Test SVA | RMA of Second Test SVA |
|------|----------------------|-----------------------|------------------------|
| 1    | 8000 0000 0000 0000  | 000001                | 000000                 |
| 1    | 8000 0020 0000 0010  | 000402                | 000401                 |
| 1    | 8000 0000 0000 0000  | 000003                | 000002                 |
| 1    | 8000 0020 0000 0010  | 000400                | 000403                 |
| 2    | 8000 0000 0000 0000  | 000005                | 000004                 |
| 2    | 8000 0020 0000 0010  | 000406                | 000405                 |
| 2    | 8000 0000 0000 0000  | 000007                | 000006                 |
| 2    | 8000 0020 0000 0010  | 000404                | 000407                 |
| 3    | 8000 0000 0000 0000  | 000009                | 000008                 |
| 3    | 8000 0020 0000 0010  | 00040A                | 000409                 |
| 3    | 8000 0000 0000 0000  | 00000B                | 00000A                 |
| 3    | 8000 0020 0000 0010  | 000408                | 00040B                 |
|      |                      |                       |                        |
| v    | v                    | v                     | v                      |
| 100  | 8000 0000 03C0 0000  | 0003FD                | 0003FC                 |
| 100  | 8000 0020 03C0 0010  | 0007FE                | 0007FD                 |
| 100  | 8000 0000 03C0 0000  | 0003FF                | 0003FE                 |
| 100  | 8000 0020 03C0 0010  | 0007FC                | 0007FF                 |

80000285 A

LMT3/P - 380

Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| C      | FAKECM Allocate Cache.                     |
| E      | Write Word to SVA.                         |
| F      | Purge Cache All.                           |
| 1A     | Double Read from two SVAs; 11/11 Requests. |

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 0.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX), xxx = 0 through FFF(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Cache Set 0.
- Select all Map Sets.
- Set Cache Block Number to 0.
- Set Pass Count to 1.

Conditions:

- C.0 If Repeat Iteration selected,
- a. Reload initial value of variables.
  - b. Purge all cache.
  - c. Read and check CPU status.
- Save variables in case of repeat.
- Pass count (1 through 100(HEX).
- a. Report pass count; informational.
- C.1 Allocate current Cache block, report SVA
- a. Select FAKECM mode.
  - b. Set RF2 to 0200 0000 xxx0(HEX); SVA, xxx = current Cache block.
  - c. Start processor at micrand C, FAKECM Allocate of Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report SVA; informational.

80000285 A

LMT3/P - 381

Conditions:

- C.2 Initialize block, report write data.
- Set RFA to F0--F0(HEX); word 0 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - Set RF2 to allocate SVA plus 8.
  - Set RFA to F1--F1(HEX); word 1 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - Set RF2 to allocate SVA plus 10(HEX).
  - Set RFA to F2--F2(HEX); word 2 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - Set RF2 to allocate SVA plus 18(HEX).
  - Set RFA to F3--F3(HEX); word 3 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report F3--F3(HEX); write data; informational.
- C.3 Result of first read, Word 0 test.
- Deselect FAKECM mode.
  - Set RF2 to current first SVA.
  - Set RF3 to current second SVA.
  - Set RF6 to DF--DF(HEX); filler.
  - Set RF7 to DF--DF(HEX); filler.
  - Set CM; 1000(HEX) to current PTD entry.
  - Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 1A, double read from two SVAs 11/11 requests, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.4 Result of second read, Word 0 test.
- Read RF7, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.

60000285 A

LMT3/P - 382

Conditions:

- C.5 Result of first read, Word 1 test.
- Set RF2 to current first SVA.
  - Set RF3 to current second SVA.
  - Set RF6 to DF--DF(HEX); filler.
  - Set RF7 to DF--DF(HEX); filler.
  - Select TF(1), set bit 40 in processor PTM register.
  - Set CM 1000(HEX) to current PTD entry.
  - Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 1A, double read from two SVAs 11/11 requests, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.6 Result of second read, Word 1 test.
- Read RF7, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.7 Result of first read, Word 2 test.
- Set RF2 to current first SVA.
  - Set RF3 to current second SVA.
  - Set RF6 to DF--DF(HEX); filler.
  - Set RF7 to DF--DF(HEX); filler.
  - Select TF(2), set bit 40 and 41 in processor PTM register.
  - Set CM 1000(HEX) to current PTD entry.
  - Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 1A, double read from different SVAs 11/11 requests, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.8 Result of second read, Word 2 test.
- Read RF7, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.9 Result of first read, Word 3 test.
- Set RF2 to current first SVA.
  - Set RF3 to current second SVA.
  - Set RF6 to DF--DF(HEX); filler.
  - Set RF7 to DF--DF(HEX); filler.
  - Select TF(3), set bit 40, 41 and 42 in processor PTM register.
  - Set CM; 1000(HEX), to current PTD entry.
  - Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 1A, double read from different SVAs 11/11 requests, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.

60000285 A

LMT3/P - 383

Conditions (C.9 continued):

- k. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.

C.10 Result of second read, Word 3 test.

- a. Read RF7, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.

C.11 CPU Status Word 1.

- a. Report word 1 if saved; error if reported.

C.12 CPU Status Word 2.

- a. Report word 2 if saved; error if reported.

C.13 First Failure Capture Information.

- a. Report PFS87 bits 19-23 in bits 59-63; information only.
- b. Report last PTL written in bits 34-47; information only.
- c. Report last PTA written in bits 13-31; information only.

Clear bits 40, 41, and 42 in processor PTM register.

Exit Subsection if 100(HEX) passes complete; else advance Cache Block Number and return to C.0.

#### Section 34, Subsection 1

This subsection tests the Hit Not Valid logic, the Cache Tag Word Address Valid RAM, and the Cache Tag File Address RAM in Cache Set 1.

#### Hardware Tested:

- Word Address Valid RAMs.
- Tag File Address RAMs. Z
- Hit Not Valid Logic. Z
- Tag File Address (TFADPE) Parity Checker. Z
- Cache Block Address (CABRP) Parity Generator. A

#### Method:

Conditions 1 and 2 are used to initialize the current Cache Block; that is, the block to be tested. The data is set to known constants and the Tag set so a miss is ensured.

Two conditions are required to test one word. Therefore conditions 3 and 4 test word 0, conditions 5 and 6 test word 1, etc.

Conditions 3, 5, 7, and 9 perform the actual test, the double read, then report the result of the first read.

Conditions 4, 6, 8, and 10 report the result of the second or Hit Not Valid read.

This subsection issues a Byte Load as its second request. This tests the SAWOI gates when the Tag File is updated.

For additional detail see the section description.

Test Patterns (hexadecimal):

| Pass | Cache Block         | Word Tested    | SVA of First Read |      |      | SVA of Second Read |      |      |
|------|---------------------|----------------|-------------------|------|------|--------------------|------|------|
| 1    | 0                   | 0              | 0000              | 0000 | 0008 | 0000               | 0000 | 0000 |
| 1    | 0                   | 1              | 0000              | 0100 | 0010 | 0000               | 0100 | 0008 |
| 1    | 0                   | 2              | 0000              | 0000 | 0018 | 0000               | 0000 | 0010 |
| 1    | 0                   | 3              | 0000              | 0100 | 0000 | 0000               | 0100 | 0018 |
| 2    | 1                   | 0              | 0000              | 0000 | 0028 | 0000               | 0000 | 0020 |
| 2    | 1                   | 1              | 0000              | 0100 | 0030 | 0000               | 0100 | 0028 |
| 2    | 1                   | 2              | 0000              | 0000 | 0038 | 0000               | 0000 | 0030 |
| 2    | 1                   | 3              | 0000              | 0100 | 0020 | 0000               | 0100 | 0038 |
| 3    | 2                   | 0              | 0000              | 0000 | 0048 | 0000               | 0000 | 0040 |
| 3    | 2                   | 1              | 0000              | 0100 | 0050 | 0000               | 0100 | 0048 |
| 3    | 2                   | 2              | 0000              | 0000 | 0058 | 0000               | 0000 | 0050 |
| 3    | 2                   | 3              | 0000              | 0100 | 0040 | 0000               | 0100 | 0058 |
|      |                     |                |                   |      |      |                    |      |      |
| v    | v                   | v              | v                 |      |      | v                  |      |      |
| 100  | FF                  | 0              | 0000              | 0000 | 1FE8 | 0000               | 0000 | 1FE0 |
| 100  | FF                  | 1              | 0000              | 0100 | 1FF0 | 0000               | 0100 | 1FE8 |
| 100  | FF                  | 2              | 0000              | 0000 | 1FF8 | 0000               | 0000 | 1FF0 |
| 100  | FF                  | 3              | 0000              | 0100 | 1FE0 | 0000               | 0100 | 1FF8 |
|      |                     |                | RMA of First      |      |      | RMA of Second      |      |      |
| Pass | PTD Entry (CM 1000) |                | Test SVA          |      |      | Test SVA           |      |      |
| 1    | B000                | 0000 0000 0000 | 000001            |      |      | 000000             |      |      |
| 1    | B000                | 0020 0000 0010 | 000402            |      |      | 000401             |      |      |
| 1    | B000                | 0000 0000 0000 | 000003            |      |      | 000002             |      |      |
| 1    | B000                | 0020 0000 0010 | 000400            |      |      | 000403             |      |      |
| 2    | B000                | 0000 0000 0000 | 000005            |      |      | 000004             |      |      |
| 2    | B000                | 0020 0000 0010 | 000408            |      |      | 000405             |      |      |
| 2    | B000                | 0000 0000 0000 | 000007            |      |      | 000006             |      |      |
| 2    | B000                | 0020 0000 0010 | 000404            |      |      | 000407             |      |      |

| Pass | PTD Entry (CM 1000) | RMA of First<br>Test SVA | RMA of Second<br>Test SVA |
|------|---------------------|--------------------------|---------------------------|
| 3    | B000 0000 0000 0000 | 000009                   | 000008                    |
| 3    | B000 0020 0000 0010 | 00040A                   | 000409                    |
| 3    | B000 0000 0000 0000 | 000008                   | 00000A                    |
| 3    | B000 0020 0000 0010 | 000408                   | 00040B                    |
| ↓    |                     |                          |                           |
| 100  | B000 0000 03C0 0000 | 0003FD                   | 0003FC                    |
| 100  | B000 0020 03C0 0010 | 0007FE                   | 0007FD                    |
| 100  | B000 0000 03C0 0000 | 0003FF                   | 0003FE                    |
| 100  | B000 0020 03C0 0010 | 0007FC                   | 0007FF                    |

#### Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| C      | FAKECM Allocate Cache.                     |
| E      | Write Word to SVA.                         |
| F      | Purge Cache All.                           |
| 1B     | Double Read from Two SVAs; IF/SA Requests. |

#### Initialization:

- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set P Descriptor register = 0.
- Select Cache Set 1.
- Set RF12 to 7; byte length, eight bytes.
- Set Cache Block Number to 0.
- Set Pass Count to 1.

#### Conditions:

- C.0 If Repeat Iteration selected,
- a. Reload initial value of variables.
  - b. Purge all cache.
  - c. Read and check CPU status.

Save variables in case of repeat.

- Pass count (1 through 100(HEX).
- a. Report pass count; informational.

#### Conditions:

- C.1 Allocate current Cache block, report SVA.
- a. Select FAKECM mode.
  - b. Set RF2 to 0200 0000 xxx0(HEX); SVA, xxx = current Cache block.
  - c. Start processor at micrand C, FAKECM Allocate of Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report SVA; informational.
- C.2 Initialize block, report write data.
- a. Set RFA to F0--F0(HEX); word 0 write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - d. Set RF2 to allocate SVA plus 8.
  - e. Set RFA to F1--F1(HEX); word 1 write data.
  - f. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - g. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - h. Set RF2 to allocate SVA plus 10(HEX)
  - i. Set RFA to F2--F2(HEX); word 2 write data.
  - j. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - k. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - l. Set RF2 to allocate SVA plus 18(HEX).
  - m. Set RFA to F3--F3(HEX); word 3 write data.
  - n. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - o. Read CPU status, format and save if error; error if not 0.
  - p. Report F3--F3; write data; informational.
- C.3 Result of first read, Word 0 test.
- a. Deselect FAKECM mode.
  - b. Set RF10 to current first SVA.
  - c. Set RF2 to current second SVA.
  - d. Set RF14 to DF--DF(HEX); filler.
  - e. Set RF6 to DF--DF(HEX); filler.
  - f. Set CM; 1000(HEX) to current PTD entry.
  - g. Set CM at first RMA to A5--A5(HEX).
  - h. Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - i. Read CPU status, format and save if error; error if not 0.
  - j. Start processor at micrand 1B, double read from two SVAs IF/SA requests, wait for CPU halted.
  - k. Read CPU status, format and save if error; error if not 0.
  - l. Read RF14, report data; error if not A5--A5(HEX).
- C.4 Result of second read, Word 0 test.
- a. Restore original data (C3--C3000xxx(HEX) to CM at first RMA.
  - b. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.

Conditions:

- C.5 Result of first read, Word 1 test.  
a. Set RF10 to current first SVA.  
b. Set RF2 to current second SVA.  
c. Set RF14 to DF--DF(HEX); filler.  
d. Set RF6 to DF--DF(HEX); filler.  
e. Select TF(1), set bit 40 in processor PTM register.  
f. Set CM 1000(HEX) to current PTD entry.  
g. Set CM at first RMA to A5--A5(HEX).  
h. Start processor at micrand F, Purge All Cache, wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Start processor at micrand 1B, double read from two SVAs IF/SA requests, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Read RF14, report data; error if not A5--A5(HEX).
- C.6 Result of second read, Word 1 test.  
a. Restore original data; C3--C3000xxx(HEX), to CM at first RMA.  
b. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.7 Result of first read, Word 2 test.  
a. Set RF10 to current first SVA.  
b. Set RF2 to current second SVA.  
c. Set RF14 to DF--DF(HEX); filler.  
d. Set RF6 to DF--DF(HEX); filler.  
e. Select TF(2), set bit 40 and 41 in processor PTM register.  
f. Set CM; 1000(HEX) to current PTD entry.  
g. Set CM at first RMA to A5--A5(HEX).  
h. Start processor at micrand F, Purge All Cache, wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Start processor at micrand 1B, double read from different SVAs IF/SA requests, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Read RF14, report data; error if not A5--A5(HEX).
- C.8 Result of second read, Word 2 test.  
a. Restore original data; C3--C3000xxx(HEX) to CM at first RMA.  
b. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.9 Result of first read, Word 3 test.  
a. Set RF10 to current first SVA.  
b. Set RF2 to current second SVA.  
c. Set RF14 to DF--DF(HEX); filler.  
d. Set RF6 to DF--DF(HEX); filler.  
e. Select TF(3), set bits 40, 41 and 42 in processor PTM register.  
f. Set CM; 1000(HEX) to current PTD entry.  
g. Set CM at first RMA to A5--A5(HEX).  
h. Start processor at micrand F, Purge All Cache, wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Start processor at micrand 1B, double read from different SVAs IF/SA requests, wait for CPU halted.

Conditions (C.9 continued):

- k. Read CPU status, format and save if error; error if not 0.  
l. Read RF14, report data; error if not A5--A5(HEX).
- C.10 Result of second read, Word 3 test.  
a. Restore original data; C3--C3000xxx(HEX), to CM at first RMA.  
b. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.11 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Clear bits 40, 41, and 42 in processor PTM register.

Exit Subsection if 100(HEX) passes complete; else advance Cache Block Number and return to C.0.

Section 34, Subsection 2

This subsection tests the Hit Not Valid logic, the Cache Tag Word Address Valid RAM, and the Cache Tag File Address RAM in Cache Set 2.

Hardware Tested:

- Word Address Valid RAMs.
- Tag File Address RAMs.
- Hit Not Valid Logic.
- Tag File Address (TFADPE) Parity Checker.
- Cache Block Address (CABRP) Parity Generator.

Method:

Conditions 1 and 2 are used to initialize the current Cache Block; that is, the block to be tested. The data is set to known constants and the Tag set so a miss is ensured.

Two conditions are required to test one word. Therefore conditions 3 and 4 test word 0, conditions 5 and 6 test word 1, etc.

Conditions 3, 5, 7, and 9 perform the actual test, the double read, then report the result of the first read.

Conditions 4, 6, 8, and 10 report the result of the second or Hit Not Valid read.

This subsection issues a IF request as its second request. This tests the Instr Fetch Word of Interest (IFWOI) gates when the Tag File is updated.

For additional detail see the section description.

Test Patterns (hexadecimal):

| Pass | Cache Block | Word Tested | SVA of First Read | SVA of Second Read |
|------|-------------|-------------|-------------------|--------------------|
| 1    | 0           | 0           | 0000 0000 0008    | 0000 0000 0000     |
| 1    | 0           | 1           | 0000 0100 0010    | 0000 0100 0008     |
| 1    | 0           | 2           | 0000 0000 0018    | 0000 0000 0010     |
| 1    | 0           | 3           | 0000 0100 0000    | 0000 0100 0008     |
| 2    | 1           | 0           | 0000 0000 0028    | 0000 0000 0020     |
| 2    | 1           | 1           | 0000 0100 0030    | 0000 0100 0028     |
| 2    | 1           | 2           | 0000 0000 0038    | 0000 0000 0030     |
| 2    | 1           | 3           | 0000 0100 0020    | 0000 0100 0038     |
| 3    | 2           | 0           | 0000 0000 0048    | 0000 0000 0040     |
| 3    | 2           | 1           | 0000 0100 0050    | 0000 0100 0048     |
| 3    | 2           | 2           | 0000 0000 0058    | 0000 0000 0050     |
| 3    | 2           | 3           | 0000 0100 0040    | 0000 0100 0058     |
|      |             |             |                   |                    |
| v    | v           | v           | v                 | v                  |
| 100  | FF          | 0           | 0000 0000 1FE8    | 0000 0000 1FE0     |
| 100  | FF          | 1           | 0000 0100 1FF0    | 0000 0100 1FE8     |
| 100  | FF          | 2           | 0000 0000 1FF8    | 0000 0000 1FF0     |
| 100  | FF          | 3           | 0000 0100 1FE0    | 0000 0100 1FF8     |

| Pass | PTD Entry (CM 1000) | RMA of First Test SVA | RMA of Second Test SVA |
|------|---------------------|-----------------------|------------------------|
| 1    | B000 0000 0000 0000 | 000001                | 000000                 |
| 1    | B000 0020 0000 0010 | 000402                | 000401                 |
| 1    | B000 0000 0000 0000 | 000003                | 000002                 |
| 1    | B000 0020 0000 0010 | 000400                | 000403                 |
| 2    | B000 0000 0000 0000 | 000005                | 000004                 |
| 2    | B000 0020 0000 0010 | 000406                | 000405                 |
| 2    | B000 0000 0000 0000 | 000007                | 000006                 |
| 2    | B000 0020 0000 0010 | 000404                | 000407                 |

| Pass | PTD Entry (CM 1000) | RMA of First Test SVA | RMA of Second Test SVA |
|------|---------------------|-----------------------|------------------------|
| 3    | B000 0000 0000 0000 | 000009                | 000008                 |
| 3    | B000 0020 0000 0010 | 00040A                | 000409                 |
| 3    | B000 0000 0000 0000 | 00000B                | 00000A                 |
| 3    | B000 0020 0000 0010 | 000408                | 00040B                 |
|      |                     |                       |                        |
| v    | v                   | v                     | v                      |
| 100  | B000 0000 03C0 0000 | 0003FD                | 0003FC                 |
| 100  | B000 0020 03C0 0010 | 0007FE                | 0007FD                 |
| 100  | B000 0000 03C0 0000 | 0003FF                | 0003FE                 |
| 100  | B000 0020 03C0 0010 | 0007FC                | 0007FF                 |

Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| C      | FAKECM Allocate Cache.                     |
| E      | Write Word to SVA.                         |
| F      | Purge Cache All.                           |
| 1C     | Double Read from Two SVAs; II/IF requests. |

Initialization:

- Select Cache Set 2 if available else select Set 0.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FF(HEX)).
- Set Cache Block Number to 0.
- Set Pass Count to 1.

Conditions:

- C.0 If Repeat Iteration selected,
- Reload initial value of variables.
  - Purge all cache.
  - Read and check CPU status.
- Save variables in case of repeat.
- Pass count (1 through 100(HEX)).
- Report pass count; informational.

Conditions:

- C.1 Allocate current Cache block, report SVA.  
a. Select FAKECM mode.  
b. Set RF2 to 0200 0000 xxx0(HEX); SVA, xxx = current Cache block.  
c. Start processor at micrand C, FAKECM Allocate of Cache, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report SVA; informational.
- C.2 Initialize block, report write data.  
a. Set RFA to F0--F0(HEX); word 0 write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.  
d. Set RF2 to allocate SVA plus 8.  
e. Set RFA to F1--F1(HEX); word 1 write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.  
h. Set RF2 to allocate SVA plus 10(HEX).  
i. Set RFA to F2--F2(HEX); word 2 write data.  
j. Start processor at micrand E, write word to SVA, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.  
l. Set RF2 to allocate SVA plus 18(HEX).  
m. Set RFA to F3--F3(HEX); word 3 write data.  
n. Start processor at micrand E, write word to SVA, wait for CPU halted.  
o. Read CPU status, format and save if error; error if not 0.  
p. Report F3--F3(HEX); write data; informational.
- C.3 Result of first read, Word 0 test.  
a. Deselect FAKECM mode.  
b. Set RF2 to current first SVA.  
c. Set RF10 to current second SVA.  
d. Set RF8 to DF--DF(HEX); filler.  
e. Set RF14 to DF--DF(HEX); filler.  
f. Set CM 1000(HEX) to current PTD entry.  
g. Set CM at second request RMA to A5--A5(HEX).  
h. Start processor at micrand F, Purge All Cache, wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Start processor at micrand 1C, double read from two SVAs II/IF requests, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Read RF8, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.4 Result of second read, Word 0 test.  
a. Restore original data; C3--C3000xxx(HEX) at RMA of second request.  
b. Read RF14, report data; error if not A5--A5(HEX).

60000285 A

LMT3/P - 392

Conditions:

- C.5 Result of first read, Word 1 test.  
a. Set RF2 to current first SVA.  
b. Set RF10 to current second SVA.  
c. Set RF8 to DF--DF(HEX); filler.  
d. Set RF14 to DF--DF(HEX); filler.  
e. Select TF(1), set bit 40 in processor PTM register.  
f. Set CM 1000(HEX) to current PTD entry.  
g. Set CM at second request RMA to A5--A5(HEX).  
h. Start processor at micrand F, Purge All Cache, wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Start processor at micrand 1C, double read from two SVAs II/IF requests, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Read RF8, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.6 Result of second read, Word 1 test.  
a. Restore original data; C3--C3000xxx(HEX), at RMA of second request.  
b. Read RF14, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.7 Result of first read, Word 2 test.  
a. Set RF10 to current first SVA.  
b. Set RF3 to current second SVA.  
c. Set RF8 to DF--DF(HEX); filler.  
d. Set RF14 to DF--DF(HEX); filler.  
e. Select TF(2), set bit 40 and 41 in processor PTM register.  
f. Set CM 1000(HEX) to current PTD entry.  
g. Set CM at second request RMA to A5--A5(HEX).  
h. Start processor at micrand F, Purge All Cache, wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Start processor at micrand 1C, double read from different SVAs II/IF requests, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Read RF8, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.8 Result of second read, Word 2 test.  
a. Restore original data; C3--C3000xxx(HEX), at RMA of second request.  
b. Read RF14, report data; error if not A5--A5(HEX).

60000285 A

LMT3/P - 393



# Conditions:

- C.9 Result of first read, Word 3 test.
- Set RF2 to current first SVA.
  - Set RF10 to current second SVA.
  - Set RF8 to DF--DF(HEX); filler.
  - Set RF14 to DF--DF(HEX); filler.
  - Select TF(3), set bit 40, 41 and 42 in processor PTM register.
  - Set CM 1000(HEX) to current PTD entry.
  - Set CM at second request RMA to A5--A5(HEX).
  - Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 1C, double read from different SVAs II/IF requests, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.10 Result of second read, Word 3 test.
- Restore original data; C3--C3000xxx(HEX), at RMA of second request.
  - Read RF14, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.11 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Clear bits 40, 41, and 42 in processor PTM register.

Exit Subsection if 100(HEX) passes complete; else advance Cache Block Number and return to C.0.

## Section 34, Subsection 3

This subsection tests the Hit Not Valid logic, the Cache Tag Word Address Valid RAM, and the Cache Tag File Address RAM in Cache Set 3.

### Hardware Tested:

- Word Address Valid RAMs.
- Tag File Address RAMs
- Hit Not Valid Logic.
- Tag File Address (TFADPE) Parity Checker.
- Cache Block Address (CABRP) Parity Generator.

60000285 A

LMT3/P - 394

# Method:

Conditions 1 and 2 are used to initialize the current Cache Block; that is, the block to be tested. The data is set to known constants and the Tag set so a miss is ensured.

Two conditions are required to test one word. Therefore conditions 3 and 4 test word 0, conditions 5 and 6 test word 1, etc.

Conditions 3, 5, 7, and 9 perform the actual test, the double read, then report the result of the first read.

Conditions 4, 6, 8, and 10 report the result of the second or Hit Not Valid read.

For additional detail see the section description.

## Test Patterns (hexadecimal):

| Pass | Cache Block | Word Tested | SVA of First Read | SVA of Second Read |
|------|-------------|-------------|-------------------|--------------------|
| 1    | 0           | 0           | 0000 0000 0008    | 0000 0000 0000     |
| 1    | 0           | 1           | 0000 0100 0010    | 0000 0100 0008     |
| 1    | 0           | 2           | 0000 0000 0018    | 0000 0000 0010     |
| 1    | 0           | 3           | 0000 0100 0000    | 0000 0100 0018     |
| 2    | 1           | 0           | 0000 0000 0028    | 0000 0000 0020     |
| 2    | 1           | 1           | 0000 0100 0030    | 0000 0100 0028     |
| 2    | 1           | 2           | 0000 0000 0038    | 0000 0000 0030     |
| 2    | 1           | 3           | 0000 0100 0020    | 0000 0100 0038     |
| 3    | 2           | 0           | 0000 0000 0048    | 0000 0000 0040     |
| 3    | 2           | 1           | 0000 0100 0050    | 0000 0100 0048     |
| 3    | 2           | 2           | 0000 0000 0058    | 0000 0000 0050     |
| 3    | 2           | 3           | 0000 0100 0040    | 0000 0100 0058     |
| I    | I           | I           | I                 | I                  |
| V    | V           | V           | V                 | V                  |
| 100  | FF          | 0           | 0000 0000 1FE8    | 0000 0000 1FE0     |
| 100  | FF          | 1           | 0000 0100 1FF0    | 0000 0100 1FE8     |
| 100  | FF          | 2           | 0000 0000 1FF8    | 0000 0000 1FF0     |
| 100  | FF          | 3           | 0000 0100 1FE0    | 0000 0100 1FF8     |

60000285 A

LMT3/P - 395

| Pass   | PTD Entry CM 1000)  | RMA of First<br>Test SVA | RMA of Second<br>Test SVA |
|--------|---------------------|--------------------------|---------------------------|
| 1      | B000 0000 0000 0000 | 000001                   | 000000                    |
| 1      | B000 0020 0000 0010 | 000402                   | 000401                    |
| 1      | B000 0000 0000 0000 | 000003                   | 000002                    |
| 1      | B000 0020 0000 0010 | 000400                   | 000403                    |
| 2      | B000 0000 0000 0000 | 000005                   | 000004                    |
| 2      | B000 0020 0000 0010 | 000406                   | 000405                    |
| 2      | B000 0000 0000 0000 | 000007                   | 000006                    |
| 2      | B000 0020 0000 0010 | 000404                   | 000407                    |
| 3      | B000 0000 0000 0000 | 000009                   | 000008                    |
| 3      | B000 0020 0000 0010 | 00040A                   | 000409                    |
| 3      | B000 0000 0000 0000 | 00000B                   | 00000A                    |
| 3      | B000 0020 0000 0010 | 000408                   | 00040B                    |
| I<br>V | I<br>V              | I<br>V                   | I<br>V                    |
| 100    | B000 0000 03C0 0000 | 0003FD                   | 0003FC                    |
| 100    | B000 0020 03C0 0010 | 0007FE                   | 0007FD                    |
| 100    | B000 0000 03C0 0000 | 0003FF                   | 0003FE                    |
| 100    | B000 0020 03C0 0010 | 0007FC                   | 0007FF                    |

#### Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| C      | FAKECM Allocate Cache.                     |
| E      | Write Word to SVA.                         |
| F      | Purge Cache All.                           |
| IA     | Double Read from Two SVAs; II/II requests. |

#### Initialization:

- Select Cache Set 3 if available; else select set 1.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX), xxx = 0 through FFF(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set Current Cache Block to 0.
- Set Pass Count to 1.

#### Conditions:

- C.0 If Repeat Iteration selected,
- a. Reload initial value of variables.
  - b. Purge all cache.
  - c. Read and check CPU status.
- Save variables in case of repeat.
- Pass count (1 through 100(HEX).
- a. Report pass count; informational.
- C.1 Allocate current Cache block, report SVA.
- a. Select FAKECM mode.
  - b. Set RF2 to 0200 0000 xxx0(HEX); SVA, xxx = current Cache block.
  - c. Start processor at micrand C, FAKECM Allocate of Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report SVA; informational.
- C.2 Initialize block, report write data.
- a. Set RFA to F0--F0(HEX); word 0 write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - d. Set RF2 to allocate SVA plus 8.
  - e. Set RFA to F1--F1(HEX); word 1 write data.
  - f. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - g. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - h. Set RF2 to allocate SVA plus 10(HEX).
  - i. Set RFA to F2--F2(HEX); word 2 write data.
  - j. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - k. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - l. Set RF2 to allocate SVA plus 18(HEX).
  - m. Set RFA to F3--F3(HEX); word 3 write data.
  - n. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - o. Read CPU status, format and save if error; error if not 0.
  - p. Report F3--F3(HEX); write data; informational.
- C.3 Result of first read, Word 0 test.
- a. Deselect FAKECM mode.
  - b. Set RF2 to current first SVA.
  - c. Set RF3 to current second SVA.
  - d. Set RF8 to DF--DF(HEX); filler.
  - e. Set RF7 to DF--DF(HEX); filler.
  - f. Set CM 1000(HEX) to current PTD entry.
  - g. Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.

Conditions (C.3 continued):

- i. Start processor at micrand 1A, double read from two SVAs II/II requests, wait for CPU halted.
  - j. Read CPU status, format and save if error; error if not 0.
  - k. Read RF6, report data; error if not C3--C3000xxx(HEX), xxx = current RMA for first SVA.
- C.4 Result of second read, Word 0 test.
  - a. Read RF7, report data; error if not C3--C3000xxx(HEX), xxx = current RMA for second SVA.
- C.5 Result of first read, Word 1 test.
  - a. Set RF2 to current first SVA.
  - b. Set RF3 to current second SVA.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Set RF7 to DF--DF(HEX); filler.
  - e. Select TF(1), set bit 40 in processor PTM register.
  - f. Set CM 1000(HEX) to current PTD entry.
  - g. Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Start processor at micrand 1A, double read from two SVAs II/II requests, wait for CPU halted.
  - j. Read CPU status, format and save if error; error if not 0.
  - k. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.6 Result of second read, Word 1 test.
  - a. Read RF7, report data; error if not C3--C3000xxx(HEX), xxx = current RMA for second SVA.
- C.7 Result of first read, Word 2 test.
  - a. Set RF2 to current first SVA.
  - b. Set RF3 to current second SVA.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Set RF7 to DF--DF(HEX); filler.
  - e. Select TF(2), set bits 40 and 41 in processor PTM register.
  - f. Set CM 1000(HEX) to current PTD entry.
  - g. Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Start processor at micrand 1A, double read from different SVAs II/II requests, wait for CPU halted.
  - j. Read CPU status, format and save if error; error if not 0.
  - k. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.8 Result of second read, Word 2 test.
  - a. Read RF7, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.

Conditions:

- C.9 Result of first read, Word 3 test.
  - a. Set RF2 to current first SVA.
  - b. Set RF3 to current second SVA.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Set RF7 to DF--DF(HEX); filler.
  - e. Select TF(3), set bit 40, 41 and 42 in processor PTM register.
  - f. Set CM 1000(HEX) to current PTD entry.
  - g. Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Start processor at micrand 1A, double read from different SVAs II/II requests, wait for CPU halted.
  - j. Read CPU status, format and save if error; error if not 0.
  - k. Read RF6, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for first SVA.
- C.10 Result of second read, Word 3 test.
  - a. Read RF7, report data; error if not C3--C3000xxx(HEX); xxx = current RMA for second SVA.
- C.11 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.12 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.13 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Clear bits 40, 41, and 42 in processor PTM register.

Exit Subsection if 100(HEX) passes complete; else advance Cache Block Number and return to C.0.

## SECTION 35 - CMC TAG DECODER RANK A/RANK B LOGIC AND DATA

Test the Rank A Response Network. Use Instruction Fetch (IF) and Address Control A-Stream (SA) requests to generate the test responses.

Test the Rank B Response Network in a similar manner. Also test the Rank B Comparator and the Rank B Data Latch.

Subsection 0 - CMC Tag Decoder Rank A Response Network.

Subsection 1 - CMC Tag Decoder Rank B Logic and Data, Part 1.

Subsection 2 - CMC Tag Decoder Rank B Logic and Data, Part 2.

### Method:

Refer to subsection descriptions.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 0.
- Select all Map Sets.

## Section 35, Subsection 0

Test Rank A Response Network of the CMC Tag Decoder.

This subsection tests the Rank A Comparator and the response signals from LM to Instruction Issue, the Instruction Fetch unit and the Address Control A-Stream (SA) when a Rank A Compare condition occurs.

### Hardware Tested:

- IF/SA Rank A Responses.
- IF/SA Update Tag File logic.

### Method:

The Rank A logic is tested by issuing two read requests to the same Cache block. The block in Cache is first initialized by writing a known pattern in all four words. When the test micrand sequence is started, the first read request causes a block fill to start, the second read request is timed to result in a Rank A compare.

Condition 2 tests the II response and the Rank A 2 bit Comparator. A different word is compared during each one of the four passes.

60000285 A

LMT3/P - 400

Condition 6 tests the Stream-A response and Condition 10 tests the Instruction Fetch response.

A failure would either hang waiting for a response, or transfer the old data to the Register File.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description                               |
|--------|-------------------------------------------|
| C      | FAKECM Allocate Cache.                    |
| E      | Write Word to SVA.                        |
| 10     | Set P Descriptor register.                |
| 1A     | Double Read from two SVAs II/II requests. |
| 1B     | Double Read from two SVAs IF/SA requests. |
| 1C     | Double Read from two SVAs II/IF requests. |

### Initialization:

- Set P Descriptor register to 0--0.
  - Set RF11 to 0--0; segment number.
  - Set CM 0 to 0--0; P Descriptor value.
  - Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Select Cache Set 0 only.
- Set Pass Count to 1.

### Conditions:

- C.0 Pass count (1 through 4).
- If iteration count = (1, 4, 7, A), go to C.5.
  - If iteration count = (2, 5, 8, B), go to C.9.
  - Report pass count; informational.
- C.1 Allocate Cache block 0 for II test, report SVA.
- Select FAKECM mode.
  - Set RF2 to 0200 0000 0000(HEX) (SVA).
  - Start processor at micrand C, FAKECM Allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.

60000285 A

LMT3/P - 401

Conditions:

- C.2 Initialize block, report write data.
- Set RFA to F0--F0(HEX); word 0 write data
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error, report write data and exit condition.
  - Set RF2 to 0200 0000 0008(HEX); SVA, word 1.
  - Set RFA to F1--F1(HEX); word 1 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - Set RF2 to 0200 0000 0010(HEX); SVA, word 2.
  - Set RFA to F2--F2(HEX); word 2 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.

NOTE

Disregard skipped conditions on error display.

- Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - Set RF2 to 0200 0000 0018(HEX); SVA, word 3.
  - Set RFA to F3--F3(HEX); word 3 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report F3--F3(HEX); write data; informational.
- C.3 Result of first read; II/II requests.
- Deselect FAKECM mode.
  - Set RF2 to 0, 8, 10, 18 for passes 1, 2, 3 and 4 respectively SVA for first read.
  - Set RF3 to 8, 10, 18, 0 for passes 1, 2, 3 and 4 respectively SVA for second read.
  - Set CM 1000(HEX) to B0--0(HEX) entry.
  - Set RF8 to DF--DF(HEX); filler.
  - Set RF7 to DF--DF(HEX); filler.
  - Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand IA, double read from two SVAs; II/II requests, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not C3--C300000x(HEX); x = 0, 1, 2, 3 for passes 1, 2, 3 and 4 respectively.

80000285 A

LMT3/P - 402

Conditions:

- C.4 Result of second read.
- Read RF7, report data; error if not C3--C300000x(HEX); x = 1, 2, 3, 0 for passes 1, 2, 3 and 4 respectively.

If error, repeat iteration or stop on end iteration, go to C.13.

- C.5 Allocate Cache block 0 for SA test, report SVA.
- Select FAKECM mode.
  - Set RF2 to 0200 0000 0000(HEX) (SVA).
  - Start processor at micrand C, FAKECM Allocate Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report SVA; informational.
- C.6 Initialize block, report write data.
- Set RFA to F0--F0(HEX); word 0 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error, report write data and exit condition.
  - Set RF2 to 0200 0000 0008(HEX); SVA, word 1.
  - Set RFA to F1--F1(HEX); word 1 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - Set RF2 to 0200 0000 0010(HEX); SVA, word 2.
  - Set RFA to F2--F2(HEX); word 2 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.
  - Set RF2 to 0200 0000 0018(HEX); SVA, word 3.
  - Set RFA to F3--F3(HEX); word 3 write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report F3--F3(HEX); write data; informational.
- C.7 Result of first read; IF/SA requests.
- Deselect FAKECM mode.
  - Set RF10 to 0--0(HEX); PYA for first read, word 0.
  - Set RF2 to 0--0(HEX); SVA for second read, word 0.
  - Set RF12 to 7; byte length, eight bytes.
  - Set RF8 to DF--DF(HEX); filler.
  - Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand F, Purge All Cache, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand IB, double read from two SVAs; IF/SA requests, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not A5--A5 C3C3(HEX).

80000285 A

LMT3/P - 403

Conditions:

- C.8 Result of second read.  
a. Read RF8, report data; error if not C3--C3000000(HEX).
- If error, repeat iteration or stop on end of iteration, go to C.13.
- C.9 Allocate Cache block 0 for IF test, report SVA.  
a. Select FAKECM mode.  
b. Set RF2 to 0200 0000 0000(HEX) (SVA).  
c. Start processor at micrand C, FAKECM Allocate Cache, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report SVA; informational.
- C.10 Initialize block, report write data.  
a. Set RFA to F0--F0(HEX); word 0 write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0. If error, report write data and exit condition.  
d. Set RF2 to 0200 0000 0008(HEX); SVA, word 1.  
e. Set RFA to F1--F1(HEX); word 1 write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.  
h. Set RF2 to 0200 0000 0010(HEX); SVA, word 2.  
i. Set RFA to F2--F2(HEX); word 2 write data.  
j. Start processor at micrand E, write word to SVA, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0. If error report write data and exit condition.  
l. Set RF2 to 0200 0000 0018(HEX); SVA, word 3.  
m. Set RFA to F3--F3(HEX); word 3 write data.  
n. Start processor at micrand E, write word to SVA, wait for CPU halted.  
o. Read CPU status, format and save if error; error if not 0.  
p. Report F3--F3(HEX); write data; informational.
- C.11 Result of first read; I1/IF requests.  
a. Deselect FAKECM mode.  
b. Set RF2 to 0--08(HEX); SVA for first read, word 1.  
c. Set RF10 to 0--0(HEX); PVA for second read, word 0.  
d. Set RF8 to DF--DF(HEX); filler.  
e. Set RF14 to DF--DF(HEX); filler.  
f. Start processor at micrand F, Purge All Cache, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Start processor at micrand IC, double read from two SVAs; I1/IF Requests, wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Read RF8, report data; error if not C3--C3000001(HEX).
- C.12 Result of second read.  
a. Read RF14, report data; error if not A5--A5C3C3(HEX).

60000285 A

LMT3/P - 404

Conditions:

- C.13 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.15 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 4 passes complete; else return to C.0.

Section 35, Subsection 1

This subsection tests the CMC TAG DECODER Rank B circuitry. This includes the Rank B CMC Tag Comparator (word 0, word 1), the Rank B timing, the data latch for Rank B time and two of the three responses (I1 and SA). Part 2 tests the remaining response (IF), and the word 2, word 3 compare.

Rank B logic is used when a word fetch results in a Hit Not Valid situation and the requested word has already arrived from CM and is detected (by Tag comparison) to be in the Rank B Data Latch.

Hardware Tested:

- Rank B Comparator.
- Rank B Data Latch.

Method:

The double read method, used in Subsection 0, is used again in this subsection. However, one difference causes Rank B instead of Rank A to be used. In the previous subsection the second read requested the fourth or last word from the incoming block. In this subsection the first word is again requested. This changes the timing such that the Rank B logic is invoked.

To make sure the Rank B network is being used the micrand sequence is timed. Each micrand sequence executed is repeated 16 times by looping within the sequence. The total time is reported and must fall within 10 percent of the expected time. Timing is taken by reading the CMC Free Running Counter before and after the test sequence. However, if a Rank B miss compare should occur, no response will be generated and the CPU will likely hang.

This subsection uses the I1/I1 double read sequence and the IF/SA sequence. Each loop executes two sequences or four micrands. It is necessary to change the Cache Tag for each sequence to initiate a Block Fill. Therefore, two sequences using different SVAs must be used during each loop.

60000285 A

LMT3/P - 405

Conditions 0, 1, 2 and 3 use a 0 pattern with the II/II sequence.

Conditions 4, 5, 6 and 7 use a 1's pattern with the IF/SA sequence.

Conditions 8, 9, 10 and 11 use a 55 through 55 pattern with the II/II sequence.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                                                    |
|--------|----------------------------------------------------------------|
| D      | Read Word from SVA.                                            |
| 1D     | Double Read from two SVAs; II/II Requests.<br>Repeat 16 times. |
| 1E     | Double Read from two SVAs; IF/SA Requests.<br>Repeat 16 times. |

#### Initialization:

- Set CM 1000(HEX) to F000 0000 0000 0000(HEX); PTD entry.
- Set CM 1001(HEX) to B000 0000 0000 0000(HEX); PTD entry.
- Purge Cache A11
- Set PSM to 7F(HEX).
- Select Cache Set 0 only.
- Set RF12 to 0--07(HEX); byte length, eight bytes.

#### Conditions:

##### NOTE

Disregard skipped conditions on error display.

- C.0 If iteration count = 1, go to C.4.  
If iteration count = 2, go to C.8.  
Fill Cache block 0 for II test, report read data.
- a. Set RF2 to 0000 1000 0000(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Set CM; 0 through 3 to C3--C300000x; where x = 0 through 3.
  - g. Read RF6, report data; error if not F0--F0(HEX).

60000285 A

LMT3/P - 406

#### Conditions:

- C.1 Test Rank B II logic and Word 0 compare.
- a. Set RF2 to 0--0(HEX); first micrand SVA, word 0.
  - b. Set RF3 to 0--0(HEX); second micrand SVA, word 0.
  - c. Set RF4 to 0--01000 0000(HEX); first micrand SVA, word 0.
  - d. Set RF5 to 0--01000 0000(HEX); second micrand SVA, word 0.
  - e. Set RF6 to DF--DF(HEX); filler.
  - f. Set RF7 to DF--DF(HEX); filler.
  - g. Set RF8 to DF--DF(HEX); filler.
  - h. Set RF9 to DF--DF(HEX); filler.
  - i. Set CM 0 to 0--0; Rank B data pattern.
  - j. Clear the CMC FRC.
  - k. Start processor at micrand 1D, double read from different SVAs; II/II requests, repeat 16 times, wait for CPU halted.
  - l. Read CPU status, format and save if error; error if not 0.
  - m. Read RF6, report data; error if not 00--00.
- C.2 Result of second read, 0 pattern.
- a. Read RF7, report data; error if not 0--0.
- C.3 Rank B timing.
- a. Report the time, in CPU major cycles, from the Start processor until the CPU Halted status. Error if not within 10 percent of 2p (HEX).

If error, repeat iteration or stop on end iteration, go to C.12.

- C.4 Fill Cache block 0 for SA test, report read data.
- a. Set RF2 to 0000 1000 0000(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Set CM; 0 through 3 to F x --F (xhex); where x = 0 through 3.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Set CM; 0 through 3 to C3--C300000x; where x = 0 through 3.
  - g. Read RF6, report data; error if not F0--F0(HEX).
- C.5 Test Rank B SA logic.
- a. Set RF10 to 0--0; first micrand PVA, word 0.
  - b. Set RF2 to 0--0; second micrand SVA, word 0.
  - c. Set RFF to 0--01000 0000(HEX); first micrand PVA, word 0.
  - d. Set RF3 to 0--01000 0000(HEX); second micrand SVA, word 0.
  - e. Set RF6 to DF--DF(HEX); filler.
  - f. Set RF7 to DF--DF(HEX); filler.
  - g. Set RF14 to DF--DF(HEX); filler.
  - h. Set CM 0 to FF--FF(HEX); Rank B pattern.
  - i. Clear the CMC FRC.
  - j. Start processor at micrand 1E, double read from two SVAs; IF/SA requests repeat 16 times, wait for CPU halted.
  - k. Read CPU status, format and save if error; error if not 0.
  - l. Read RF14, report data; error if not A5--A5FFFF(HEX).
- C.6 Result of second read, ones pattern.
- a. Read RF6, report data; error if not FFFF FFFF FFFF FFFF(HEX).

60000285 B

LMT3/P - 407

# Conditions:

- C.7 Rank B timing.
- Report the time, in CPU major cycles, from the Start processor until the CPU Halted status. Error if not within 10 percent of 32 (HEX).

If error, repeat iteration or stop on end iteration, go to C.12.

- C.8 Fill Cache block 0 for II test, report read data.
- Set RF2 to 0000 1000 0000(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set CM; 0 through 3 to C3--C300000x; where x = 0 through 3.
  - Read RF8, report data; error if not F0--F0(HEX).

- C.9 Test Rank B II logic and Word 1 compare.
- Set RF2 to 0--08(HEX); first micrand SVA, word 1.
  - Set RF3 to 0--08(HEX); second micrand SVA, word 1.
  - Set RF4 to 0--01000 0008(HEX); first micrand SVA, word 1.
  - Set RF5 to 0--01000 0008(HEX); second micrand SVA, word 1.
  - Set RF6 to DF--DF(HEX); filler.
  - Set RF7 to DF--DF(HEX); filler.
  - Set RF8 to DF--DF(HEX); filler.
  - Set RF9 to DF--DF(HEX); filler.
  - Set CM 1 to 55--55(HEX); Rank B pattern.
  - Clear the CMC FRC.
  - Start processor at micrand 1D, double read from two SVAs; II/II requests, repeat 16 times, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 55--55(HEX).

- C.10 Result of second read, 55's pattern.
- Read RF7, report data; error if not 55--55(HEX).

- C.11 Rank B timing.
- Report the time, in CPU major cycles, from the Start processor until the CPU Halted status. Error if not within 10 percent of 28 (HEX).

- C.12 CPU Status Word 1.
- Report word 1 if saved; error if reported.

- C.13 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.14 First Failure Capture Information.
- Report PFSB7 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

## Section 35, Subsection 2

This subsection tests the CMC TAG DECODER Rank B circuitry. This includes the Rank B CMC Tag Comparator (word 2, word 3), the Rank B timing, the data latch for Rank B time and two of the three responses (II and IF). Part 1 tested the SA response, and the word 0, word 1 compare.

Rank B logic is used when a word fetch results in a Hit Not Valid situation and the requested word has already arrived from CM and is detected (by Tag comparison) to be in the Rank B Data Latch.

### Hardware Tested:

- CMC TAG DECODER Rank B Comparator.
- Rank B Data Latch.

### Method:

The double read method, used in Subsection 0, is used again in this subsection. However, one difference causes Rank B instead of Rank A to be used. In the previous subsection the second read requested the fourth (last) word from the incoming block. In this subsection during the II/II requests the first word is requested, and during the II/IF requests the third word is requested. This changes the timing such that the Rank B logic is invoked.

To make sure the Rank B network is being used the micrand sequence is timed. Each micrand sequence executed is repeated 16 times by looping within the sequence. The total time is reported and must fall within 10 percent of the expected time. Timing is taken by reading the CMC Free Running Counter before and after the test sequence. However, if a Rank B miss compare should occur, no response will be generated and the CPU will likely hang.

This subsection uses the II/II double read sequence and the II/IF sequence. Each loop executes two sequences or four micrands. It is necessary to change the Cache Tag for each sequence to initiate a Block Fill. Therefore, two sequences using different SVAs must be used during each loop.

Conditions 0, 1, 2 and 3 use a 0 pattern with the II/II sequence.

Conditions 4, 5, 6 and 7 use a 1's pattern with the II/IF sequence.

Conditions 8, 9, 10 and 11 use a AA-AA pattern with the II/II sequence.

### Test Patterns:

Refer to Conditions.



# Micrands Used:

| Number | Description                                                    |
|--------|----------------------------------------------------------------|
| D      | Read Word from SVA.                                            |
| ID     | Double Read from two SVAs; II/II Requests.<br>Repeat 16 times. |
| IF     | Double Read from two SVAs; II/IF Requests.<br>Repeat 16 times. |

## Initialization:

- Set CM 1000(HEX) to F000 0000 0000 0000(HEX); PTD entry.
- Set CM 1001(HEX) to B000 0000 1000 0000(HEX); PTD entry.
- Purge Cache All.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

C.0 If iteration count = 1, go to C.4.

If iteration count = 2, go to C.8.

Fill Cache block 0 for II test, report read data.

- a. Set RF2 to 0--01000 0000(HEX) (SVA).
- b. Set RF6 to DF--DF(HEX); filler.
- c. Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Set CM; 0 through 3 to C3--C300000x(HEX); where x = 0 through 3.
- g. Read RF6, report data; error if not F0--F0(HEX).

C.1 Test Rank B II logic and Word 2 compare.

- a. Set RF2 to 0--010(HEX); first micrand SVA, word 2.
- b. Set RF3 to 0--010(HEX); second micrand SVA, word 2.
- c. Set RF4 to 0--01000 0010(HEX); first micrand SVA, word 2.
- d. Set RF5 to 0--01000 0010(HEX); second micrand SVA, word 2.
- e. Set RF6 to DF--DF(HEX); filler.
- f. Set RF7 to DF--DF(HEX); filler.
- g. Set RF8 to DF--DF(HEX); filler.
- h. Set RF9 to DF--DF(HEX); filler.
- i. Set CM 2 to 0--0 Rank B data pattern.
- j. Clear the CMC FRC.
- k. Start processor at micrand ID, double read from different SVAs; II/II requests, repeat 16 times, wait for CPU halted.
- l. Read CPU status, format and save if error; error if not 0.
- m. Read RF6, report data; error if not 0--0.

60000285 A

LMT3/P - 410

## Conditions:

C.2 Result of second read, 0 pattern.

- a. Read RF7, report data; error if not 0--0.

C.3 Rank B timing.

- a. Report the time, in CPU major cycles, from the Start processor until the CPU Halted status. Error if not within 10 percent of 2A (HEX).

If error, repeat iteration or stop on end iteration, go to C.12.

C.4 Fill Cache block 0 for IF test, report read data.

- a. Set RF2 to 0--01000 0000(HEX) (SVA).
- b. Set RF6 to DF--DF(HEX); filler.
- c. Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Set CM; 0 through 3 to C3--C300000x(HEX); where x = 0 through 3.
- g. Read RF6, report data; error if not F0--F0(HEX).

C.5 Test Rank B IF logic.

- a. Set RF2 to 0--0; first micrand SVA, word 0.
- b. Set RF10 to 0--010(HEX); second micrand SVA, word 2.
- c. Set RF3 to 0--01000 0000(HEX); first micrand SVA, word 0.
- d. Set RFF to 0--01000 0010(HEX); second micrand SVA, word 2.
- e. Set RF6 to DF--DF(HEX); filler.
- f. Set RF14 to DF--DF(HEX); filler.
- g. Set RF7 to DF--DF(HEX); filler.
- h. Set CM 2 to FF--FF(HEX); Rank B pattern.
- i. Clear the CMC FRC.
- j. Start processor at micrand IF, double read from two SVAs; II/IF requests, repeat 16 times, wait for CPU halted.
- k. Read CPU status, format and save if error; error if not 0.
- l. Read RF6, report data; error if not C3--C3000000(HEX).

C.6 Result of second read, 1's pattern.

- a. Read RF14, report data; error if not A5--A5 FFFF(HEX).

C.7 Rank B timing.

- a. Report the time, in CPU major cycles, from the Start processor until the CPU Halted status. Error if not within 10 percent of 3E (HEX).

If error, repeat iteration or stop on end iteration, go to C.12.

C.8 Fill Cache block 0 for II test, report read data.

- a. Set RF2 to 0--01000 0000(HEX) (SVA).
- b. Set RF6 to DF--DF(HEX); filler.
- c. Set CM; 0 through 3 to Fx--Fx(HEX); where x = 0 through 3.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Set CM; 0 through 3 to C3--C300000x(HEX); where x = 0 through 3.
- g. Read RF6, report data; error if not F0--F0(HEX).

60000285 B

LMT3/P - 411

Conditions:

- C.9 Test Rank B II logic and Word 3 compare.
- a. Set RF2 to 0--018(HEX); first micrand SVA, word 3.
  - b. Set RF3 to 0--018(HEX); second micrand SVA, word 3.
  - c. Set RF4 to 0--01000 0018(HEX); first micrand SVA, word 3.
  - d. Set RF5 to 0--01000 0018(HEX); second micrand SVA, word 3.
  - e. Set RF6 to DF--DF(HEX); filler.
  - f. Set RF7 to DF--DF(HEX); filler.
  - g. Set RF8 to DF--DF(HEX); filler.
  - h. Set RF9 to DF--DF(HEX); filler.
  - i. Set CM 3 to AA--AA(HEX); Rank B pattern.
  - j. Clear the CMC FRC.
  - k. Start processor at micrand 10, double read from two SVAs; II/II requests, repeat 16 times, wait for CPU halted.
  - l. Read CPU status, format and save if error; error if not 0.
  - m. Read RF6, report data; error if not AA--AA(HEX).
- C.10 Result of second read, AA's pattern.
- a. Read RF7, report data; error if not AA--AA(HEX).
- C.11 Rank B timing.
- a. Report the time, in CPU major cycles, from the Start processor until the CPU Halted status. Error if not within 10 percent of 2A (HEX).
- C.12 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.13 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.14 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

SECTION 36 - FAKECM AND FAULT RESPONSES

Test the Response Network in LM and the Response Signals transmitted from LM to Instruction Fetch (IFRSP) and Address Control A-Stream (SARSP). These signals are tested during FAKECM mode operations and when Page Table Faults are sensed.

- Subsection 0 - FAKECM Responses.
- Subsection 1 - Page Table Fault Responses.
- Subsection 2 - Page Faults with Double Response and Test Operations.
- Subsection 3 - Special Index/Test Situations.
- Subsection 4 - Blocking of write request to CMU.

Method:

Refer to subsection methods.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).

Section 36, Subsection 0

Test FAKECM responses.

This subsection tests the generation and transmission of the Response Signals to Instr Fetch (IF) and the Address Control A Stream (SA) when the CPU is in FAKECM mode.

Hardware Tested:

- Response Network.

Method:

FAKECM mode is selected and Cache location zero, in Set 0, is allocated and written.

Location zero is then read using a II Request (used and tested before, but repeated here to provide comparison data) a SA Request (a byte read of eight bytes, issued from Address Control), and a IF Request (set the P register to a PVA).

In the case of the II Request and the SA Request the data read acts as proof the response was received. However, in the IF Request situation, the word fetched from CM is used to address a micrand in Control Store. If the correct micrand is executed, Register File location (RF14) is set to the constant A5A5 A5A5 C3C3(HEX). This is proof the response was received.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                |
|--------|----------------------------|
| C      | FAKECM Allocate Cache.     |
| D      | Read Word from SVA.        |
| E      | Write Word to SVA.         |
| 10     | Set P Descriptor register. |
| 13     | Set P register; start IF.  |
| 19     | Read Byte from SVA.        |

#### Initialization:

- Set P Descriptor register to 0--0.
  - Set RF11 to 0--0; segment number.
  - Set CM 0 to 0--0; P Descriptor value.
  - Start processor at micrand 10, set P Descriptor, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Select Cache Set 0 only.

#### Conditions:

- C.0 Allocate Cache address 0, report SVA.
- a. Set RF2 to 0--0 (SVA).
  - b. Start processor at micrand C, FAKECM Allocate Cache, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report SVA; informational.
- C.1 Write data to Cache address 0, report write data.
- a. Set RFA to C3C3 0000 C3C3 0000(HEX); write data.
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Report write data; informational.

60000285 A

LMT3/P - 414

#### Conditions:

- C.2 Read Cache address 0, IIRSP, report read data.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not as reported in C.1.
- C.3 Read Cache address 0, SARSP, report read data.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Set RF12 to 0--07(HEX); byte length, eight bytes.
  - c. Start processor at micrand 19, read byte from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not as reported in C.1.
- C.4 Read Cache address 0, IFRSP, report read Data.
- a. Set RF14 to DF--DF(HEX); filler.
  - b. Set RF10 to 0--0; byte address.
  - c. Start processor at micrand 13, set P register; start IF, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF14, report data; error if not A5--A5 C3C3(HEX).
- C.5 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.6 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.7 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

#### Section 36, Subsection 1

#### Test Page Table Fault responses.

This subsection tests the generation of the Response Signals sent to Instruction Fetch (IFRSP) and to the Address Control A-Stream (SARSP) when a Page Table Fault Condition occurs.

#### Hardware Tested:

- Page Table Fault Response.
- Suppress Page Table Fault.

60000285 A

LMT3/P - 415

#### Method:

To create the Page Table Fault condition, a single Page Table Descriptor (PTD) is set up at CM 1000(HEX) that has the Continue bit clear and the Tag set so a match will not occur. A data fetch micrand is issued (II, SA, or IF) and the CPU status sensed. The Page Search Without Find (PSWF) status should be set. A second micrand, exactly the same as the first, is issued using a PTD which will result in a match. The second micrand confirms that the response from the first was received.

The Suppress PSWF logic is tested by selecting bit 27 in the LM Functional Unit Micrand (IISUP) and issuing the same II request. The CPU status should not indicate a PSWF condition.

#### NOTE

Suppress PSWF cannot be selected during IF operations.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                        |
|--------|------------------------------------|
| D      | Read Word from SVA.                |
| 19     | Read Bytes from SVA.               |
| 13     | Set P register; start IF.          |
| 21     | Read Word from SVA; Suppress PSWF. |
| 12     | Purge Map All.                     |

#### Initialization:

- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 through FF(HEX) to C3--C30000xx(HEX); xx = 0 through FF(HEX).
- Set RF2 to 0--0 (SVA).

60000285 A

LMT3/P - 416

#### Conditions:

#### NOTE

Disregard skipped conditions on error display.

- C.0 If iteration count = 1, go to C.3.  
If iteration count = 2, go to C.6.  
If iteration count = 3, go to C.9.  
Test IIRSP with PSWF, report CPU Status Word 1.  
a. Set CM 1000(HEX) to B010 0000 0000 0000(HEX); PTD entry for miss.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU Status, format and report word 1; error if not 0.
- C.1 CPU Status Word 2.  
a. Report word 2; error if not 0000040--0(HEX) (PSWF).
- C.2 Issue read to check for CPU hang.  
a. Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry for match.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C3000000(HEX).
- If error, repeat iteration or stop on end iteration, go to C.12.
- C.3 Test SARSP with PSWF, report CPU Status Word 1.  
a. Set RF12 to 7; byte length, eight bytes.  
b. Set CM 1000(HEX) to B010 0000 0000 0000(HEX); PTD entry for miss.  
c. Start processor at micrand 19, read bytes from SVA, wait for CPU halted.  
d. Read CPU Status, format and report word 1; error if not 0.
- C.4 CPU Status Word 2.  
a. Report word 2; error if not 0000040--0(HEX); PSWF.
- C.5 Issue read to check for CPU hang.  
a. Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry for match.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand 19, read bytes from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C3000000(HEX).

60000285 A

LMT3/P - 417

Conditions:

If error, repeat iteration or stop on end iteration, go to C.12.

- C.6 Test IFRSP with PSWF, report CPU Status Word 1.
- Set RF10 to 0--0; P register value.
  - Set RF14 to DF--DF(HEX); filler.
  - Set CM 1000(HEX) to B010 0000 0000 0000(HEX); PTD entry for miss.
  - Start processor at micrand 13, Set P register; start IF, wait for CPU halted.
  - Read CPU Status, format and report word 1; error if not 0.
- C.7 CPU Status Word 2,
- Report word 2; error if not 0000440--0(HEX); Abnormal Halt and PSWF.
- C.8 Issue IF to check for CPU hang.
- Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry for match.
  - Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand 13, Set P register; start IF, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not A5--A5 C3C3(HEX).
- If error, repeat iteration or stop on end iteration, go to C.12.
- C.9 Test suppress PSWF, report CPU Status Word 1.
- Set CM 1000(HEX) to B010 0000 0000 0000(HEX); PTD entry for miss.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand 21, read word from SVA; Suppress PSWF, wait for CPU halted.
  - Read CPU status, format and report word 1; error if not 0.
- C.10 CPU Status Word 2.
- Report word 2; error if not 0.
- C.11 Issue read, check for CPU hang.
- Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry for match.
  - Set RF6 to DF--DF; filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C3000000(HEX).
- C.12 CPU Status Word 1, from Hang Checks.
- Report word 1 if saved; error if reported.
- C.13 CPU Status Word 2, from Hang Checks.
- Report word 2 if saved; error if reported.

60000285 A

LMT3/P - 418

Conditions:

- C.14 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Section 36, Subsection 2

Test Page Faults with double response and test operations.

This subsection tests the response generation logic when Page Fault conditions occur during a Test Operation and a Test and Set Bit Operation.

Hardware Tested:

- Page Map Control.

Method:

Three areas in the response logic are tested by this subsection.

- During a Test Operation a Page Map miss, Page Table hit occurs. Prior to this subsection, the test has always used the Page Map. Condition 0 purges the Map, sets a PTD into CM and issues a Test Operation. A Page Table hit is expected; a RMA and count of zero should be returned. Conditions 1 and 2 report the CPU Status; zero is expected.
- During a Test Operation a Page Map miss, Page Table miss occurs resulting in a Page Search Without Find (PSWF). Condition 3 clears the PTD in CM and purges the Page Map. The Test Operation data should show no RMA (0) and the PSWF flag set. Conditions 4 and 5 report the CPU Status; PSWF is expected.
- Condition 6 tests that during a Test and Set Bit Operation a PSWF occurs. This test is intended to check the Page Fault logic when a Double Response instruction is issued. Condition 6 uses the same SVA as the previous two tests (0) and the bit to set is zero. The Register File location used to receive the test bit from CM should not be written due to the PSWF condition. Conditions 7 and 8 report the CPU Status; PSWF is expected.

Test Patterns:

Refer to Conditions.

60000285 A

LMT3/P - 419

# Micrands Used:

| Number | Description       |
|--------|-------------------|
| F      | Purge Cache All.  |
| 12     | Purge Map All.    |
| 15     | Test Operation.   |
| 20     | Test and Set Bit. |

## Initialization:

- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU Status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU Status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 1000(HEX) to B0--0(HEX) (PTD entry).

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 If iteration count = 1, go to C.3.  
 If iteration count = 2, go to C.8.  
 Check Test response after Page Table Hit.  
 a. Set RF2 to 0--0 (SVA).  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand 15, Test, wait for CPU halted.  
 d. Read RF8, report data; error if not 0--0.
- C.1 Report CPU Status Word 1.  
 a. Read CPU Status, format and report Status Word 1; error if not 0.
- C.2 Report CPU Status Word 2.  
 b. Format and report Status Word 2; error if not 0.

If error, repeat iteration or stop on end iteration, go to C.8.

## Conditions:

- C.3 Check Test response after page fault.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Clear CM 1000(HEX), PTD entry.  
 c. Start processor at micrand 12, Purge Map, wait for CPU halted.  
 d. Start processor at micrand 15, Test, wait for CPU halted.  
 e. Read RF8, report data; error if not 0--080000000(HEX).
- C.4 Report CPU Status Word 1.  
 a. Read CPU Status, format and report Status Word 1; error if not 0.
- C.5 Report CPU Status Word 2.  
 a. Format and report Status Word 2; error if not 0000040-0(HEX) (PSWF).
- If error, repeat iteration or stop on end iteration, go to C.8.
- C.6 Test PSWF after Test and set bit, report CPU Status Word 1.  
 a. Set RFA to 0--0; bit number.  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand 20, Test and Set Bit, wait for CPU halted.  
 d. Read CPU Status, format and report Status Word 1; error if not 0.
- C.7 Report CPU Status Word 2.  
 a. Format and report Status Word 2; error if not 0000040-0(HEX) (PSWF).
- C.8 Report CPU Status Word 1 from initialization purges.  
 a. Report Word 1 if saved; error if reported.
- C.9 Report CPU Status Word 2 from initialization purges.  
 a. Report Word 2 if saved; error if reported.
- C.10 First Failure Capture Information.  
 a. Report PFS87 bits 19-23 in bits 59-63; information only.  
 b. Report last PTL written in bits 34-47; information only.  
 c. Report last PTA written in bits 13-31; information only.

## Section 36, Subsection 3

Test the unique situations where RMA with Index and RMA with Test cause specific hardware functions to be initiated.

## Hardware Tested:

- Inhibit Mark Lines.
- RMA/Test Response.

#### Method:

Condition 0 is used to test the RMA/Index function. When a CM write is issued using RMA mode addressing and the Index micrand bit is also selected, bytes 0 and 1 are not written to CM. This function transfers only bits 16 through 63 to CM. It is tested by performing the write and checking CM for bytes 0 and 1 equal zero, and bytes 2 through 7 equal to the data written.

Conditions 1, 2, and 3 test the RMA/Test function. When a prevalidate operation is selected in RMA mode and the Test micrand bit is selected, a response, via the Test Response logic is sent in reply. This situation is tested by issuing the proper micrand, waiting for CPU halted and reporting CPU Status. A second micrand (read CM using RMA) is issued to ensure the CPU is not hung up.

Conditions 4 and 5 report CPU status if a status error is detected during Conditions 0 or 3.

Test Patterns: Refer to Conditions.

| Micrands Used: | Number | Description                      |
|----------------|--------|----------------------------------|
|                | 1      | Read CM using RMA.               |
|                | 3      | Write CM using RMA with Index.   |
|                | 9      | Prevalidate using RMA with Test. |

Initialization: - None.

#### Conditions:

- C.0 Check six-byte write operation; RMA/Index.
- Set RF2 to 0--0 (RMA).
  - Set RFA to F0F1F2F3F4F5F6F7(HEX); write data.
  - Set CM 0 to DF--DF(HEX); filler.
  - Start processor at micrand 3, Write CM using RMA with Index, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not DFDF2F3F4F5F6F7(HEX).
- C.1 Check Test response after prevalidate with RMA.
- Start processor at micrand 9, Prevalidate using RMA with Test, wait for CPU halted.
  - Read CPU Status, format and report Word 1; error if not 0.
- C.2 Report Status Word 2.
- Format and report Status Word 2; error if not 0.
- C.3 Issue read, check for CPU hang.
- Set RF6 to DF--DF(HEX); filler.
  - Set CM 0 to F0F1F2F3F4F5F6F7(HEX); read data.
  - Start processor at micrand 1, Read CM using RMA, wait for CPU halted.
  - Read CPU Status, format and save if error; error if not 0.
  - Read RF6, report data; error if not F0F1F2F3F4F5F6F7(HEX).

60000265 A

LMT3/P - 422

#### Conditions:

- C.4 CPU Status Word 1.
- Report Word 1 if saved; error if reported.
- C.5 CPU Status Word 2.
- Report Word 2 if saved; error if reported.
- C.6 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

#### Section 36, Subsection 4

Test the unique situation where a Central Memory access is blocked when memory is constantly busy.

Hardware Tested: - The access/reply logic within Central Memory.

#### Method:

Micrand sequence 2B(HEX), which writes to location 0 in CM 32 times, will be started. The first write is all zeros; each time you write to CM, the write data is incremented by one. While the micrand sequence is being performed, the IOU attempts a read of CM location 0. The data read by the IOU will indicate whether the IOU had its request blocked or not. If the request was blocked, the data read will be 1F(HEX), which indicates that the read could not be performed until the micrand sequence was completed and stopped.

Due to the erratic timing between the IOU and the CPU, up to five attempts will be made at the procedure within condition 0; the test will look for one good pass. If an error is detected in all five attempts at condition C.0, the final attempt is the trial which is reported as an error.

| Micrands Used: | Number | Description                                                                |
|----------------|--------|----------------------------------------------------------------------------|
|                | 2B     | Write to CM 32 times with an incrementing data pattern starting with zero. |

#### Conditions:

- C.0 Check access by IOU during CM operation.
- Set RF2 to 0-0 (RMA for CM loc 0).
  - Set CM loc 0 to DF--DF(HEX); filler.
  - Master clear P3 and clear P3 and M3 errors.
  - Start micrand sequence 2B(HEX); do not wait for halted.
  - Read CM location 0; report error if data read does not equal 0--0X (HEX) where X is equal to a value of 2 through 1C(HEX), and all five attempts were made.
  - If all five attempts have not yet been attempted and if an error has been detected, go to step 6 and repeat the procedure.
- C.1 Report CPU Status Word 1.
- Read CPU Status, format and report Status Word 1; error if not 0.

60000265 B

LMT3/P - 423

**Conditions:**

**C.2 Report CPU Status Word 2.**

- a. Format and report Status Word 2; error if not 0.

**C.3 First Failure Capture Information.**

- a. Report PFS87 bits 19-23 in bits 59-63; information only.
- b. Report last PTL written in bits 34-47; information only.
- c. Report last PTA written in bits 13-31; information only.

THIS PAGE HAS INTENTIONALLY BEEN  
LEFT BLANK DUE TO A LATE CHANGE



## SECTION 37 - CM INVALIDATE/CACHE BYPASS

This section tests the CM Invalidate Operation and the Cache Purge and Bypass select. Both operations perform similar functions. The CM Invalidate will purge a 4-word block from Cache if there is a Tag match, and write a 60-bit word to CM. The IOU write instruction must be a C170 write; i.e., five 12-bit bytes. The Cache Purge and Bypass performs the same function, except the address is submitted from the CPU and the write data is 64 bits.

Cache Bypass is also tested by using all the methods which can invoke the Cache Bypass feature.

Subsection 0 - CM Invalidate and Cache Purge Four Word Block.

Subsection 1 - Cache Bypass.

Subsection 2 - CM Invalidate address bits 40, 41.

Method:

Refer to subsection descriptions.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).

Section 37, Subsection 0

The CM Invalidate (CMIVAD) and the Purge Bypass operations provide similar functions. Both are tested here to enhance isolation.

Hardware Tested:

- Address Mux A CMIVAD bits 40 through 47.
- AMUXA CMIVAD bits 48 through 60.
- Purge/Invalidate Control.
- Block Purge (BLKPUR) Signal to each Set.

Method:

The CM Invalidate purges one Cache block (4 words) from an IOU command. The Purge/Bypass does the same thing from a CPU or microcode command. These two operations when used together can aid in reducing suspect hardware.

60000285 A

LMT3/P - 424

One pass through the test conditions allocates and writes to the same test block in two sets using a zero's pattern. The data is read back to prove its existence then purged in set 0 using CM Invalidate, and read again to prove it was purged. The sequence is repeated in set 1 using Purge/Bypass. A ones pattern is now used in a similar manner except the Purge/Bypass is used first to test the BLKPUR Signal to set 0. Also, the second Purge/Bypass is issued from Address Control. This allows testing of the Cache Purge and Bypass Select OR gate.

A second pass is required to complete the test using alternate bit patterns. The second pass will use sets 2 and 3 if available.

If the CM Invalidate fails and the Purge/Bypass does not in both patterns of the same pass, the problem is likely an Invalidate Control problem. If only the invalidate to one pattern fails, it is probably a bit problem in AMUXA, and so on.

Test Patterns (hexadecimal):

| Pass | Conditions    | Set    | SVAs             | Type     |
|------|---------------|--------|------------------|----------|
| 1    | 1 through 3   | 0      | FFFF 0000 0000   | Inv.     |
| 1    | 4 through 6   | 1      | FFFF 0000 0000** | P/B (11) |
| 1    | 7 through 9   | 0      | FFFF 00FF FFF8** | P/B (AC) |
| 1    | 10 through 12 | 1      | FFFF 00FF FFF8** | Inv.     |
| 2    | 1 through 3   | 2 or 0 | FFFF 0055 5550** | Inv.     |
| 2    | 4 through 6   | 3 or 1 | FFFF 0055 5550** | P/B (11) |
| 2    | 7 through 9   | 2 or 0 | FFFF 00AA AAA8** | P/B (AC) |
| 2    | 10 through 12 | 3 or 1 | FFFF 00AA AAA8** | Inv.     |

\*\*Addresses are shown as configured for a 16 megabyte CM system. All variables are masked to the CM configuration under test.

Write data = C3xx yyyy yyyy(HEX).

xx = Cache set number.  
yy = Current SVA.

Micrands Used:

| Number | Description            |
|--------|------------------------|
| C      | FAKECM Allocate Cache. |
| D      | Read word from SVA.    |
| E      | Write word from SVA.   |
| F      | Purge Cache All.       |
| 20     | Test and Set Bit.      |
| 22     | Cache Purge/Bypass.    |

60000285 A

LMT3/P - 425

# Initialization:

- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set condition number to -1(FE(HEX)).
- Select FAKECM mode.
- Set all (4) PTDs into CM.
  - CM 11FE(HEX) = BFFF F000 000--0(HEX).
  - CM 1000(HEX) = BFFF F01F FFC0--0(HEX).\*\*
  - CM 10AA(HEX) = BFFF F00A AA80--0(HEX).\*\*
  - CM 1154(HEX) = BFFF F015 5540--0(HEX).\*\*
- Set Pass Count to 1.

# Conditions:

## NOTE

Disregard skipped conditions on error display.

- C.0 If iteration count = (1,5), go to C.4.  
 If iteration count = (2,6), go to C.7.  
 If iteration count = (3,7), go to C.10.  
 If running on CPU-1, go to C.4.  
 Pass count (1 or 2).  
 a. Report pass count; informational.
- C.1 Allocate and write Cache, report write data.  
 a. Select Cache Set 0; pass 1, Set 2 if available; pass 2.  
 b. Set RF2 to current SVA.  
 c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Set RFA to current write data.  
 f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 g. Read CPU status, format and save if error; error if not 0.  
 h. Report write data.
- C.2 Report read data before invalidate.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not as reported in C.1.

60000285 A

LMT3/P - 426

# Conditions:

- C.3 Report read data after invalidate.  
 a. Deselect FAKECM mode.  
 b. Invalidate CM; pass 1 - CM 0.  
     pass 2 - CM AAAAAA(HEX).\*\*  
 c. Select FAKECM mode.  
 d. Set RF8 to DF--DF(HEX); filler.  
 e. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 f. Read CPU status, format and save if error; error if not 0.  
 g. Read RF8, report data; error if not 0.
- If error, repeat iteration or stop on end iteration, go to C.13.
- C.4 Allocate and write Cache, report write data.  
 a. Select Cache Set 1; pass 1, Set 3 if available; pass 2.  
 b. Set RF2 to current SVA.  
 c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Set RFA to current write data.  
 f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
 g. Read CPU status, format and save if error; error if not 0.  
 h. Report write data.
- C.5 Report read data before purge.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not as reported in C.4.
- C.6 Report read data after purge/bypass (II).  
 a. Deselect FAKECM mode.  
 b. Start processor at micrand 22, Purge/Bypass, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Select FAKECM mode.  
 e. Set RF8 to DF--DF(HEX); filler.  
 f. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 g. Read CPU status, format and save if error; error if not 0.  
 h. Read RF8, report data; error if not 0.
- If error, repeat iteration or stop on end iteration, go to C.13.

\*\*Addresses are shown as configured for a 128 megabyte CM system. All variables are masked to the CM configuration under test.

60000285 A

LMT3/P - 427

Conditions:

- C.7 Allocate and write Cache, report write data.  
a. Select Cache Set 0; pass 1, Set 2 if available; pass 2.  
b. Set RF2 to current SVA.  
c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Set RFA to current write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Report write data.
- C.8 Report read data before purge.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not as reported in C.7.
- C.9 Report read data after purge/bypass (AC).  
a. Deselect FAKECM mode.  
b. Set RFA to 0--0; bit number.  
c. Start processor at micrand 20, Test and Set Bit, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Select FAKECM mode.  
f. Set RF6 to DF--DF(HEX); filler.  
g. Start processor at micrand D, read word from SVA, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Read RF6, report data; error if not 0.

If error, repeat iteration or stop on end iteration, go to C.13.

- C.10 If running on CPU-1, go to C.13.  
Allocate and write Cache, report write data.  
a. Select Cache Set 1; pass 1, Set 3 if available; pass 2.  
b. Set RF2 to current SVA.  
c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Set RFA to current write data.  
f. Start processor at micrand E, write word to SVA, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Report write data.
- C.11 Report read data before invalidate.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not as reported in C.10.

60000285 A

LMT3/P - 428

Conditions:

- C.12 Report read data after invalidate.  
a. Deselect FAKECM mode.  
b. Invalidate CM; pass 1 - CM FFFFFFFF.\*\*  
pass 2 - CM 555555).\*\*  
c. Select FAKECM mode.  
d. Set RF6 to DF--DF(HEX); filler.  
e. Start processor at micrand D, read word from SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF6, report data; error if not 0.
- C.13 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.15 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if two passes complete; else return to C.0.

Deselect FAKECM mode.

Section 37, Subsection 1

This subsection tests the five remaining methods which may be used to invoke a Cache Bypass operation.

Hardware Tested:

- Bypass Cache Control.

Method:

Bypass Cache operations have been successfully used prior to this subsection. That was before the testing of Cache was complete. This subsection can now test the various methods used to initiate a Cache Bypass and test Cache to ensure it was bypassed.

\*\*Addresses are shown as configured for a 128 megabyte CM system. All variables are masked to the CM configuration under test.

60000285 A

LMT3/P - 429

The methods tested are:

NOCACHE - All Cache sets disabled.  
 CACBY - Requested from FU Micrand bit.  
 REAL - Requested from FU Micrand bit.  
 IFBYP - P Descriptor contains bypass code.  
 TEST - Ensure a Cache Hit does not occur.

The procedure used to test each method is basically the same.

1. CM is initialized with a known pattern.
2. Cache is also initialized using different data.
3. A Write operation is issued with the method being tested invoked. This step is skipped when testing IFBYP and TEST.
4. A read operation is issued with the method being tested invoked. This reads back the data written in step 3. When using TEST, the operation is submitted and the result simply checked for no CPU hang and the correct result.
5. A second read operation is issued, this time using Cache. This proves that the previous write/read actually did bypass Cache. This step is skipped with TEST.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description                       |
|--------|-----------------------------------|
| 1      | Read Word from RMA.               |
| A      | Write Word to RMA.                |
| D      | Read Word from SVA.               |
| E      | Write Word to SVA.                |
| F      | Purge Cache All.                  |
| 10     | Set P Descriptor register.        |
| 13     | Set P register; start IF.         |
| 15     | TEST Operation.                   |
| 23     | Read Word from SVA, Cache Bypass. |
| 24     | Write Word to SVA, Cache Bypass.  |

Initialization:

- Select all Cache Sets.
- Select all Map Sets.
- Set CM 0 through FF(HEX) to C3--C300xx(HEX); xx = 0 through FF(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Fill Cache Block Zero.
  - Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry.
  - Set RF2 to 0000 0000 0000 (SVA).
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 If iteration count = 1, go to C.3.  
 If iteration count = 2, go to C.6.  
 If iteration count = 3, go to C.9.

NOCACHE, bypass write, report write data.

- a. Set CM 0 to DF--DF(HEX); filler.
- b. Set RFA to F0--F0(HEX); write data.
- c. Disable all Cache sets.
- d. Start processor at micrand E, write word to SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Read CM 0, report data; error if not F0--F0(HEX).

C.1 NOCACHE, bypass read, report read data.

- a. Set RF8 to DF--DF(HEX); filler.
- b. Start processor at micrand D, read word from SVA, wait for CPU halted.
- c. Read CPU status, format and save if error; error if not 0.
- d. Read RF8, report data; error if not F0--F0(HEX).

Conditions:

- C.2 Read Cache data, report Cache data.  
 a. Select all Cache sets.  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not C3--C30000(HEX).

If error, repeat iteration or stop on end iteration, go to C.12.

- C.3 CACBY, bypass write, report write data.  
 a. Set CM 0 to DF--DF(HEX); filler.  
 b. Start processor at micrand 24, write word to SVA; Cache Bypass, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read CM 0, report data; error if not F0--F0(HEX).
- C.4 CACBY, bypass read, report read data.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand 23, read word from SVA; Cache Bypass, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not F0--F0(HEX).

- C.5 Read Cache data, report Cache data.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not C3--C30000(HEX).

If error, repeat iteration or stop on end iteration, go to C.12.

- C.6 REAL, bypass write, report write data.  
 a. Set RF2 to 0--010(HEX); RMA, word 2.  
 b. Set RFA to F2--F2(HEX); write data.  
 c. Start processor at micrand A, write word to RMA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read CM 2, report data; error if not F2--F2(HEX).

- C.7 REAL, bypass read, report read data.  
 a. Set RF8 to DF--DF(HEX); filler.  
 b. Start processor at micrand 1, read word from RMA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF8, report data; error if not F2--F2(HEX).

Conditions:

- C.8 Read Cache data, report Cache data.  
 a. Set RF6 to DF--DF(HEX); filler.  
 b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Read RF6, report data; error if not C3--C30002(HEX).

If error, repeat iteration or stop on end iteration, go to C.12.

- C.9 IFBYP, bypass read.  
 a. Deselect all Cache sets.  
 b. Set P-Descriptor Rgtr to C0--0(HEX); invokes IFBYP.  
 c. Select all available Cache sets.  
 d. Set RF10 to 0--0; PVA with Segment = 0.  
 e. Set RF14 to DF--DF(HEX); filler.  
 f. Start processor at micrand 13, Set P register; start IF, wait for CPU halted.  
 g. Read CPU status, format and save if error; error if not 0.  
 h. Read RF14, report data; error if not A5A5 A5A5 A5A5 F0F0(HEX).

- C.10 IF Cache read.  
 a. Set CM 0 to 0--0; P Descriptor value.  
 b. Start processor at micrand 10, load P descriptor, wait for CPU halted.  
 c. Read CPU status, format and save if error; error if not 0.  
 d. Set CM 0 to F0--F0(HEX); reinitialize.  
 e. Set RF14 to DF--DF(HEX); filler.  
 f. Start processor at micrand 13, load P register; start IF, wait for CPU halted.  
 g. Read CPU status, format and save if error; error if not 0.  
 h. Read RF14, report data; error if not A5A5 A5A5 A5A5 C3C3(HEX).

- C.11 TEST, ensure bypass, report TEST data.  
 a. Set RF2 to 0--0 (SVA).  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand 15, TEST, wait for CPU halted.  
 d. Read CPU Status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not 0.

- C.12 CPU Status Word 1.  
 a. Report word 1 if saved; error if reported.

- C.13 CPU Status Word 2.  
 a. Report word 2 if saved; error if reported.

- C.14 First Failure Capture Information.  
 a. Report PFS87 bits 19-23 in bits 59-63; information only.  
 b. Report last PTL written in bits 34-47; information only.  
 c. Report last PTA written in bits 13-31; information only.

Section 37, Subsection 2

This subsection further tests the +CMIVAD bits.

Hardware Tested:

- AMUXA CMIVAD bits 40, 41.

Method:

In each pass, a CM Invalidate is issued and CPU status is checked (because nonexistent memory is also used, data will be ignored).

Test Patterns:

| Pass | Address |
|------|---------|
| 1    | 0       |
| 2    | 1FFFFFF |
| 3    | 155555  |
| 4    | 0AAAAA  |

Micrands Used:

| Number | Description      |
|--------|------------------|
| F      | Purge cache all. |

Initialization:

- Purge cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set condition number to -1(FE(HEX)).
- Set pass count to 1.

Conditions:

- C.0 Pass count (1-4).
  - a. Report pass count; informational.
- C.1 CM Invalidate and status word 1.
  - a. Test CM Invalidate.
  - b. Report status word 1.
- C.2 Status word 2.
  - a. Report status word 2.
- C.3 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if four passes are done.

## SECTION 38 - TAG FILE, ADDRESS MUX, BLOCK FILL DATA PARITY CIRCUIT

This section tests some of the parity error detection logic used during the Block Fill operation.

Subsection 0 - Tag File Parity Signal Verification (TFADPE, TAGFPE).

Subsection 1 - SELRD Status Code and Block Fill Data Parity Signal (CHCMPE) Verification.

Subsection 2 - Address Mux Parity Error Code Verification (AMUXPE).

Method:

Refer to subsection descriptions.

Initialization:

- Select all Cache Sets.
- Select all Map Sets.
- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).

### Section 38, Subsection 0

This subsection tests the status bits in the PFS82 register that indicate a parity error has occurred in the Tag File Word Count logic or Word of Interest logic (TAGFPE) or in the Tag File Cache Address logic (TFADPE).

Hardware Tested:

- TAGFPE/TFADPE Parity Logic.
- PFS82 bits 18, 19.

Method:

To test the TFADPE parity signal and status bit, a Cache address with wrong parity must be stored in the Tag File Cache Address RAM. This is done by selecting bit 26 in the processor PTM register. This forces all parity bits sent from MAC to zero. A SVA is then written to the Register File and used in a read request (Cache Miss/Block Fill). The address with wrong parity is written (by using PTM26) to the Tag File Cache Address RAM and when the Block Fill data comes in from CM to be written into Cache the Cache address is used and the parity error sensed.

60000285 A

LMT3/P - 436

The TAGFPE parity status is tested by selecting the CPTF bit in the processor PTM register. This forces the Tag File Count and Word of Interest parity bits to be complemented. A read operation (Cache Miss/Block Fill) is again used (SVA has good parity) and the TAGFPE status should set in the PFS register.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description            |
|--------|------------------------|
| D      | Read Word from SVA.    |
| C      | FAKECM Allocate Cache. |
| F      | Purge All Cache.       |

Initialization:

- Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry for read.
- Set CM; (0 through FF(HEX)), to C3--C30000xx(HEX); xx = 0 through FF(HEX).
- Purge Cache All
- Deselect all Map Sets.

Conditions:

NOTE

Disregard skipped conditions on error display.

C.0 If iteration count = 1, go to C.3.

Test TFADPE, report CPU Status Word 1.

- Select Cache Set 0.
- Set RF2 to 0000010--0100(HEX); SVA for Tag change.
- Select FAKECM mode.
- Start processor at micrand C, FAKECM allocate Cache; change Cache Tag, wait for CPU halted.
- Read CPU status, format and save if error; error if not 0.
- Deselect FAKECM mode.
- Set RF8 to DF--DF(HEX); filler.
- Set RF2 to 0--0100(HEX); SVA for test.
- Set bit 26 in processor PTM register; tag file disable parity.
- Start processor at micrand D, read word from SVA, wait for CPU halted.
- Clear bit 26.
- Read CPU status, format and report word one; error if not 000010000--0(HEX); TFADPE.

60000285 A

LMT3/P - 437

Conditions:

- C.1 CPU Status Word 2.  
a. Report word 2; error if not 80--000000(HEX); CADRPE (0).
- C.2 Read data from TFADPE test.  
a. Read RF6, report data; error if not C3--C3000020(HEX).

If error, stop on end iteration or repeat iteration, go to C.6.

- C.3 Test TAGFPE, report CPU Status Word 1.  
a. Purge all Cache  
b. Set RF2 to 0000010--0100(HEX); SVA for Tag change.  
c. Select FAKECM mode.  
d. Start processor at micrand C, FAKECM allocate Cache; change Cache Tag, wait for CPU halted.  
e. Deselect FAKECM mode.  
f. Read CPU status, format and save if error; error if not 0.  
g. Set RF2 to 0--0100(HEX); SVA for test.  
h. Set RF6 to DF--DF(HEX); filler.  
i. Set bit 27 (CPTF) in processor PTM register.  
j. Start processor at micrand D, read word from SVA, wait for CPU halted.  
k. Clear bit 27.  
l. Read CPU status, format and report word 1; error if not 000020--0 (HEX); TAGFPE.
- C.4 CPU Status Word 2.  
a. Report word 2; error if not all 0.
- C.5 Read Data from TAGFPE Test.  
a. Read RF6, report data; error if not C3--C3000020(HEX).
- C.6 CPU Status Word 1, from FAKECM allocates.  
a. Report word 1 if saved; error if reported.
- C.7 CPU Status Word 2, from FAKECM allocates.  
a. Report word 2 if saved; error if reported.
- C.8 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Repeat from C.0 if not last iteration.

Purge all Cache.

Section 38, Subsection 1

This subsection tests the four Sel Read Data (SELRD) bits (PFS82 bits 20 through 23) by selecting the different sources of data that come from LM to the Register File. This subsection also tests the CM to Cache (CHCMPE) status. This status bit indicates when the data coming from CM to Cache during a block fill operation contains wrong parity.

Hardware Tested:

- SELRD Code Transmitters.
- SELRD Code Receivers.
- CHCMPE Transmitter.
- PFS82 Bits 20 through 23 and 25.

Method:

The SELRD code is a 2-bit indication of which data source sent the 64-bit LM data to the Register File on the RDATA lines. If the incoming data is sensed to have a parity error the SELRD code is decoded in OPI and sent as four status bits to PFS82 (20 through 23). The data containing the parity error is not written to the Register File. To test this code, each situation must be uniquely created.

- Code 3 PFS82(20) - Data from the Rank B Data Latch. The same method as used to test the Rank B Data Latch is used here; a double read to the same Cache Block. The first read requests word 0, the second read requests word 2. Word 2, therefore, comes through the Rank B logic. Data containing wrong parity is set in CM word 2. This will cause a CHCMPE error signal and a SELRD code of 0 should set PFS82 bit 20. A Crampe for Cache set 0 is also expected because with PDMS enabled, the data containing parity errors coming from CM is not written (blocked) to cache. However the parity bit for the Word Valid RAM still reflects all words written.
- Code 2 PFS82(21) - RMA Information from Test or Index. Bit 22 in the processor PTM registers is set and an Index operation is issued. The parity bit for the Index search count is complemented and the wrong parity data is sent to OPI.
- Code 1 PFS82(22) - Cache Read Data after Hit. Data containing wrong parity is set into the Register File. With FAKECM mode selected, this data is written to Cache and subsequently read out. The read out should cause PFS82(22) to set.
- Code 0 PFS82(23) - Data from CM. Data containing wrong parity is written to CM. A read request is issued and a Block Fill sequence started. The word containing the wrong parity is the word requested and when received by OPI the SELRD code of 0 should be decoded and bit 23 set in the PFS82 register. A Crampe for Cache set 0 is also expected because with PDMS enabled, the data containing parity errors coming from CM is not written (blocked) to cache. However the parity bit for the Word Valid RAM still reflects all words written.



# Test Patterns:

Refer to Conditions.

## Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| C      | FAKECM Allocate Cache.                     |
| D      | Read Word from SVA.                        |
| E      | Write Word to SVA.                         |
| 15     | Test Operation.                            |
| 1A     | Double Read from Two SVAs; II/II Requests. |
| 12     | Purge Map All.                             |
| F      | Purge Cache All.                           |

## Initialization:

- Purge Page Map.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FF(HEX)).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FF(HEX)).
- Enable PDM Errors.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 If iteration count = 1, go to C.4.  
 If iteration count = 2, go to C.7.  
 If iteration count = 3, go to C.10.

Code 3 test, rank B data, result of first read.

- a. Set bits 0 and 1 in the Central Memory Environment Control register (ECR). This disables SECCED and parity checking in Central Memory.
- b. Set bit 59 in the IOU ECR. Enable IOU Test Mode register.
- c. Set CM 0 = 0--0B0(HEX); odd parity in byte seven.
- d. Set CM 1 = B1--B1(HEX); even parity.
- e. Set CM 2 = B2--B2(HEX); even parity.
- f. Set CM 3 = A3--A3(HEX); even parity.
- g. Clear bit 59 in the IOU ECR.
- h. Set RF2 to 0--0; SVA for first read, word 0.

60000285 A

LMT3/P - 440

## Conditions (C.0 continued):

- i. Set RF3 to 0--0; SVA for second read, word 0.
- j. Set RF8 to DF--DF(HEX); filler.
- k. Set RF7 to DF--DF(HEX); filler.
- l. Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry.
- m. Start processor at micrand 1A, double read from two SVAs; II/II requests, wait for CPU halted.
- n. Read RF8, report data; error if not DF--DF(HEX); Data blocked.

- C.1 Code 3 test, result of second read.  
 a. Read RF7, report data; error if not DF--DF(HEX); parity error blocks data.
- C.2 CPU Status Word 1, from Code 0 test.  
 a. Read CPU status, format and report word 1; error if not 00010940--0(HEX); CHWDPE (7), SELRD (0,3) and CHCMPE.
- C.3 CPU Status Word 2, from Code 3 test.  
 a. Report word 2 after removing the DAI 0 through 8 bits; error if not 080000010--0(HEX); CRAMPE and DAI (7).  
 b. The DAI 0 through 8 bits are removed because they are not predicted due to refresh, and so on.

If error, repeat iteration or stop on end iteration, go to C.13.

- C.4 Code 2 test, Index operation data.  
 a. Set RF2 to 0--0 (SVA).  
 b. Set RF8 to DF--DF; filler.  
 c. Set CM 1000(HEX) to 0ACBD3120--0(HEX).  
 d. Set bit 22 in processor PTM register; select complement parity to Index count.  
 e. Start processor at micrand 14, Index Operation, wait for CPU halted.  
 f. Clear bit 22 in processor PTM register.  
 g. Read RF8, report data; error if not DF--DF(HEX); parity error in DAI blocks incoming data.
- C.5 CPU Status Word 1, from Code 2 test.  
 a. Read CPU status, format and report word 1; error if not 00000200--0(HEX); SELRD 2.
- C.6 CPU Status Word 2, from Code 2 test.  
 a. Report word 2; error if not 000004080--0(HEX); PSWF and DAI (4).

If error, repeat iteration or stop on end iteration, go to C.13.

60000285 A

LMT3/P - 441

Conditions:

- C.7 Code 1 Test, Cache read data.
- Disable PDM errors.
  - Set RF2 to 0--0 (SVA).
  - Set RF6 to DF--DF(HEX); filler.
  - Set bit 18 in processor Test Mode register; select MAC data output, force parity.
  - Set RFA to 0--01(HEX); write data, byte seven contains wrong parity.
  - Clear bit 18 in processor TM register.
  - Select FAKECM mode.
  - Start processor at micrand E, write word to SVA; set bad data into Cache, wait for CPU halted.
  - Enable PDM errors.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Deselect FAKECM Mode.
  - Read RF6, report data; error if not DF--DF(HEX) (DAI parity error blocks data).
- C.8 CPU Status Word 1, from Code 1 test.
- Read CPU status, format and report word 1; error if not 0001040--0 (HEX); CHWDPE (7) and SELRD (1).
- C.9 CPU Status Word 2, from Code 1 test.
- Report word 2; error if not 0080000100100000(HEX); CHC (0), DAI (7), A-Stream (7).

If error, repeat iteration or stop on end iteration, go to C.13.

- C.10 Code 0 test, CM read data.
- Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA wait for CPU halted.
  - Read RF6, report data; error if not DF--DF(HEX); DAI parity error blocks data.
- C.11 CPU Status Word 1, from Code 0 test.
- Read CPU status, format and report word 1; error if not 00010840--0(HEX) or 00010940--0(HEX); CHWDPE (7), SELRD (0) and CHCMPE.
- C.12 CPU Status Word 2, from Code 0 test.
- Report word 2 after removing DAI 0 through 6 bits and Address Fan-in (0, 1) bits; error if not 080000010--0(HEX); CramPE and DAI (7).
  - The DAI 0 through 6 bits and Address Fan-in (0, 1) bits are removed because they are not predicted due to refresh, and so on.
- C.13 CPU Status Word 1 from purges.
- Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2 from purges.
- Report word 2 is saved; error if reported.

60000285 A

LMT3/P - 442

Conditions:

- C.15 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Clear bits 0 and 1 in Central Memory ECR.

Set CM 0 to zero (clear wrong parity).

Start processor at Micrand F, Purge All Cache.

Disable PDM errors.

Section 38, Subsection 2

This subsection tests the Address Mux PE (AMUXPE) codes that are sent from LM to OPI, decoded and sent on to PFS82 (bits 28 through 31) when a parity error is sensed in the Cache Address Register (CAR). The codes define which unit the address came to LM from.

Hardware Tested:

- AMUXPE Coder and Xmitter.
- AMUXPE Decoder.
- PFS82 bits 28 through 31.

Method:

There are four AMUXPE codes, each code must be uniquely tested.

Code 0 - Address from CM Invalidate. Word one in Cache is set to a known quantity. Bits 60 and 62 are set in the IOU Test Mode register to force parity bits to ones on CPU addresses. The CPU is started at Micrand 3C and loops there indefinitely. This allows the Invalidate PE to be sensed. A C170 PP write to address one causes CM address one to be written and Cache address one to be invalidated or purged. A parity error should be detected in the CAR and a code of 0 sent to OPI.

Code 1 - Address from Address Control. Bit 18 is set in the processor PTM register. This forces zero parity bits on data sent from MAC. An Address is set in to the Register File (contains wrong parity). A word read operation is initiated to a known Cache address hit). A parity error should be detected in CAR and a code of 1 sent to OPI.

60000285 A

LMT3/P - 443

Code 2 - Address from Instruction Fetch (IF). In a similar manner a PVA containing wrong parity is set into the Register File. This value is then set into the P Register, starting the IF sequence. A parity error should be sensed by CAR and a code of 2 sent to OPI.

Code 3 - Address from CABR (internal to LM). Using the method described in Code 1, an SVA containing wrong parity is set into the Register File. A word write is issued sending the SVA through Address Control and to the AMUX. A Code 1 error is detected. However, the address must be looped from CABR back through the AMUX to the CAR to delay for the Cache write cycle. This should also cause a Code 3 error.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                |
|--------|----------------------------|
| C      | FAKECM Allocate Cache.     |
| D      | Read Word from SVA.        |
| E      | Write Word to SVA.         |
| 10     | Set P Descriptor register. |
| 13     | Set P register; start IF.  |
| F      | Purge Cache All.           |

#### Initialization:

- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set P Descriptor Register to 0--0.
  - Set RF11 to 0--0; segment number.
  - Set CM 0 to 0--0; P Descriptor value.
  - Start processor at micrand 10, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM; 0 through FF(HEX) to C3--C30000xx(HEX); xx = 0 through FF (HEX).
- Set CM; 1000(HEX) to B0--0(HEX) (PTD for C.0).

#### Conditions:

#### NOTE

Disregard skipped conditions on error display.

60000285 A

LMT3/P - 444

#### Conditions:

- C.0 If iteration count = 1, go to C.4.  
 If iteration count = 2, go to C.7.  
 If iteration count = 3, go to C.10.

Initialize Cache block 0 (Code 0 test), and 1; Code 1 test.

- a. Set RF2 to FFFF 0000 0000(HEX); SVA for invalidate test.
- b. Select FAKECM mode.
- c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0. If error, report SVA and exit condition.
- e. Set RF2 to FFFF 0000 0010(HEX); SVA to invalidate, word 2 of block 0.
- f. Set RFA to C3C3 0000 C300 0002(HEX); write data.
- g. Start processor at micrand E, write word to SVA, wait for CPU halted.
- h. Read CPU status, format and save if error; error if not 0. If error, report write data and exit condition.
- i. Deselect FAKECM mode.
- j. Set RF2 to 0--020(HEX); SVA for Code 1 test.
- k. Set RF6 to DF--DF(HEX); filler.
- l. Start processor at micrand D, read word from SVA, wait for CPU halted.
- m. Read RF6; error if data not C3--C3000004(HEX).
- n. Read CPU status, format and save if error; error if not 0. If error (data or status) report data and exit condition.
- o. If no Ds, report 0s.

- C.1 If running on CPU-1, go to C.4.  
 Code 0 test (CMIVAD), report CPU Status Word 1.
  - a. Set bits 59 through 61 and 63 in the IOU Test Mode Register (TMR); this forces ones on address parity bits.
  - b. Set bit 59 in the IOU ECR, enable IOU TMR.
  - c. Start processor at micrand 3C, put CPU into everlasting loop, wait for CPU halted; halted will not occur, time out will occur and the CPU will appear hung.
  - d. Invalidate: Write 4567--45678 (80 bits) to CM address 2, use C170 format to invoke the invalidate.
  - e. Clear bit 59 in IOU ECR.
  - f. Reset IOU TMR to 00-01F(HEX).
  - g. Read CPU status, format and report word 1; error if not 010000080--0(HEX); CARPE (5) and AMUXPE (0).
- C.2 Code 0 test (CMIVAD), report CPU Status Word 2.
  - a. Report word 2; error if not 800780--0(HEX); CADRPE (0), CMC Error Code = 7; Address byte three PE, and CPU hung.

60000285 A

LMT3/P - 445

Conditions:

- C.3 Code 0 test (CMIVAD), report Cache data.
- Set RF2 to FFFF 0000 0010(HEX); SVA invalidated.
  - Set RF6 to DF--DF(HEX); filler.
  - Select FAKECM mode.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Deselect FAKECM mode.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0--0; data invalidated.

If error, repeat iteration or stop on end iteration, go to C.13.

- C.4 Code 1 Test (ACAD), report CPU Status Word 1.
- Set bit 18 in processor PTM register, select MAC data output to force 0 parity bits.
  - Set RF2 to 0--020(HEX); SVA with wrong parity.
  - Clear bit 18 in processor PTM register.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and report word one; error if not 010000040--0(HEX); CARPE (5) and AMUXPE (1).

- C.5 Code 1 test (ACAD), report CPU Status Word 2.
- Report word 2; error if not 80--01000000(HEX); CADRPE (set 0), SVAPE.

- C.6 Code 1 test (ACAD), report read data.
- Read RF6, report data; error if not C3--C3000004(HEX).

If error, repeat iteration or stop on end iteration, go to C.13.

- C.7 Code 2 test (IFAD), report CPU Status Word 1.
- Set bit 18 in processor PTM register, select MAC Data Output to force 0 parity bits.
  - Set RF10 to 0--010000(HEX); P register value.
  - Clear bit 18 in processor PTM register.
  - Set RF14 to DF--DF(HEX); filler.
  - Start processor at micrand 13, set P register; start IF, wait for CPU halted.
  - Set RF10 to 0--0; clear wrong parity.
  - Read CPU status, format and report word 1; error if not 040000230--0(HEX); CARPE (3), CHTGPE and AMUXPE (2, 3).

- C.8 Code 2 test (IFAD), report CPU Status Word 2.
- Report word 2; error if not 000004040--0(HEX); PSWF and DAI.

- C.9 Code 2 test (IFAD), report read data. (Data blocked due to PSWF).
- Read RF14, report data; error if not DF--DF(HEX).

If error, repeat iteration or stop on end iteration, go to C.13.

Conditions:

- C.10 Code 3 test (CAR), report CPU Status Word 1.
- Set CM 4) to DF--DF(HEX); filler.
  - Set RFA to C3--C3 0004(HEX); write data.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and report word 1; error if not 010000090--0(HEX); CARPE (5) and AMUXPE (0,3).
- C.11 Code 3 test (CAR), report CPU Status Word 2.
- Report word 2; error if not 80--01000000(HEX); CADRPE (set 0), SVAPE.
- C.12 Code 3 test (CAR), report data written.
- Read CM 4), report data; error if not C3--C3000004(HEX).
- C.13 CPU Status Word 1, from C.0/C.3.
- Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2, from C.0/C.3.
- Report word 2 if saved; error if reported.
- C.15 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Set RF2 to 0--0; clear wrong parity SVA.

Start processor at micrand 5, clear residue parity errors.

Set CM; 0 through 1FFF(HEX) to C3--C300xxxx(HEX); xxxx = 0 through 1FFF HEX).

## SECTION 39 - CACHE LOOKAHEAD, PART 1

The Cache Lookahead test is divided into three sections. This section, part 1, tests the basic feature. That is, it tests the supporting logic including the Next Block (NXTBA) incrementer and ensures the feature completes as intended.

Subsection 0 - Cache Lookahead, NXTBA to the Address mux.

Subsection 1 - Cache Lookahead, Verify Operation.

Subsection 2 - Cache Lookahead, NXTBA Incrementer.

Method:

Refer to subsection methods.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).

### Section 39, Subsection 0

This subsection tests the input to Address Mux A (AMUX) from the Next Block Address (NXTBA) Incrementer. This subsection also tests the basic Lookahead control logic.

Hardware Tested:

- NXTBA bits 32 through 47 to AMUX.
- NXTBA bits 48 through 58 to AMUX.
- Lookahead Control Logic.

Method:

The objective of this subsection is to verify bit integrity of the NXTBA input to the Address mux, and also to test the ability of LM to successfully complete a Lookahead operation.

60000285 A

LMT3/P - 448

Each Lookahead test sequence purges Cache followed by a CM read. A Cache Miss occurs and a Block Fill is initiated. This first read is intended to allocate and fill a Cache block and allocate and write the Page Map. A second read is then issued using the same address. A Cache Hit should occur and with Lookahead enabled, a Block Fill should be initiated to fill the next sequential Cache block. FAKECM mode is used to read words N and N+4 to verify the operation.

Two passes through the test conditions are required to use the four test patterns, 0s, Fs, 5s, and As.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description         |
|--------|---------------------|
| F      | Cache Purge All.    |
| D      | Read Word from SVA. |

Initialization:

- Select Unconditional Lookahead.
- Select all available Cache Sets.
- Select all Map Sets.
- Set Pass Count to 1.

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 or 2).
- a. Report pass count; informational.

If Scope Mode is set, and no error on C.1, C.2, go to C.3.

- C.1 Result from first read; Cache miss, nonlookahead.
- a. Start processor at micrand F, purge Cache all, wait for CPU halted.
  - b. Read CPU status, format and save if error; error if not 0.
  - c. Set PTD entry into CM 1000(HEX).
    - Pass 1 = B000 0000 0000 0000(HEX).
    - Pass 2 = BAAA AAAA AA80 0000(HEX).
  - d. Set RF2 to SVA, Pass 1 = 0000 0000 0000(HEX).
    - Pass 2 = AAAA 5555 5520(HEX).

60000285 A

LMT3/P - 449

Conditions (C.1 continued):

- e. Set RF8 to DF--DF(HEX); filler.
- f. Start processor at micrand D, read word from SVA, wait for CPU halted.
- g. Read CPU status, format and save if error; error if not 0.
- h. Read RF8, report data; error if not:  
Pass 1 = C3C3 C3C3 C3000000(HEX).  
Pass 2 = C3C3 C3C3 C3000024(HEX).

C.2 Result from address N+4 before lookahead.

- a. Set RF2 to SVA, Pass 1 = 0000 0000 0020(HEX).  
Pass 2 = AAAA 5555 5540(HEX).
- b. Set RF8 to DF--DF(HEX); filler.
- c. Select FAKECM mode.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Deselect FAKECM mode.
- g. Read RF8, report data; error if not 0.

If Scope Mode is set, go to C.0.  
If error on C.1, C.2, go to C.12.

C.3 If Scope Mode is set, and no error on C.3, C.4, go to C.5.

- Result from first read; Cache hit, perform lookahead.
- a. Set RF2 to SVA, Pass 1 = 0000 0000 0000(HEX).  
Pass 2 = AAAA 5555 5520(HEX).
- b. Set RF8 to DF--DF(HEX); filler.
- c. Start processor at micrand D, read word from SVA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF8, report data; error if not:  
Pass 1 = C3C3 C3C3 C3000000(HEX).  
Pass 2 = C3C3 C3C3 C3000024(HEX).

C.4 Result from address N+4 after lookahead.

- a. Set RF2 to SVA, Pass 1 = 0000 0000 0020(HEX).  
Pass 2 = AAAA 5555 5540(HEX).
- b. Set RF8 to DF--DF(HEX); filler.
- c. Select FAKECM mode.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Deselect FAKECM mode.
- g. Read RF8, report data; error if not:  
Pass 1 = C3C3 C3C3 C3000004(HEX).  
Pass 2 = C3C3 C3C3 C3000028(HEX).

If Scope Mode is set, go to C.0.  
If C.3, C.4 error, go to C.12.

Conditions:

C.5 Result from first read; Cache miss, nonlookahead.

- a. Set PTD entry into CM 1000(HEX).  
Pass 1 = BFFF FFFF FFC0 0000(HEX).  
Pass 2 = B555 5555 5540 0000(HEX).
- b. If Scope Mode is set and no error no C.5, C.6, go to C.7.
- c. Set RF2 to SVA, Pass 1 = FFFF 7FFF FFC0(HEX).  
Pass 2 = 5555 2AAA AA80(HEX).
- d. Set RF8 to DF--DF(HEX); filler.
- e. Start processor at micrand D, read word from SVA, wait for CPU halted.
- f. Read CPU status, format and save if error; error if not 0.
- g. Read RF8, report data; error if not:  
Pass 1 = C3C3 C3C3 C3000038(HEX).  
Pass 2 = C3C3 C3C3 C3000010(HEX).

C.6 Result from address N+4 before lookahead.

- a. Set RF2 to SVA, Pass 1 = FFFF 7FFF FFE0(HEX).  
Pass 2 = 5555 2AAA AAA0(HEX).
- b. Set RF8 to DF--DF(HEX); filler.
- c. Select FAKECM mode.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Deselect FAKECM mode.
- g. Read RF8, report data; error if not 0.

If Scope Mode is set, go to C.0.  
If C.5, C.6 error, go to C.12.

C.7 Result from first read; Cache hit, perform lookahead.

- a. Set RF2 to SVA, Pass 1 = FFFF 7FFF FFC0(HEX).  
Pass 2 = 5555 2AAA AA80(HEX).
- b. Set RF8 to DF--DF(HEX); filler.
- c. Start processor at micrand D, read word from SVA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF8, report data; error if not:  
Pass 1 = C3C3 C3C3 C3000038(HEX).  
Pass 2 = C3C3 C3C3 C3000010(HEX).

C.8 Result from address N+4 after lookahead.

- a. Set RF2 to SVA, Pass 1 = FFFF 7FFF FFE0(HEX).  
Pass 2 = 5555 2AAA AAA0(HEX).
- b. Set RF8 to DF--DF(HEX); filler.
- c. Select FAKECM mode.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Deselect FAKECM mode.
- g. Read RF8, report data; error if not:  
Pass 1 = C3C3 C3C3 C300003C(HEX).  
Pass 2 = C3C3 C3C3 C3000014(HEX).

#### Conditions:

If Scope Mode is set, go to C.0.

- C.9 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if two passes complete; else return to C.0.

#### Section 39, Subsection 1

This subsection completes verification of the basic Unconditional Lookahead Operation by checking all four words written to Cache.

#### Hardware Tested:

- Lookahead Control Logic.

#### Method:

Ensure the Lookahead feature functions as designed. That is, when Unconditional Lookahead is selected and a Cache Hit occurs the next sequential Cache block is filled from CM.

A Lookahead condition is created by issuing two consecutive read operations using the same SVA. After the second read all four words are read from Cache, in FAKECM mode, and verified.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                               |
|--------|-------------------------------------------|
| D      | Read Word from SVA.                       |
| F      | Purge Cache All.                          |
| 1A     | Double Read from two SVAs; II/II Request. |

60000285 A

LMT3/P - 452

#### Initialization:

- Set CM; 1000(HEX) to 8000 0000 0000 0000(HEX); PTD entry.

#### Conditions:

#### NOTE

Disregard skipped conditions on error display.

- C.0 Result of first read.
  - a. Start processor at micrand F, purge Cache, wait for CPU halted.
  - b. Read CPU status, format and save if error; error if not 0.
  - c. Set RF2 to 0--0; SVA for first read.
  - d. Set RF3 to 0--0; SVA for second read.
  - e. Set RF6 to DF--DF(HEX); filler.
  - f. Set RF7 to DF--DF(HEX); filler.
  - g. Start processor at micrand 1A, double read from two SVAs, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Read RF8, report data; error if not C3C3 C3C3 C3000000(HEX).

- C.1 Result of second read.
  - a. Read RF7, report data; error if not C3C3 C3C3 C3000000(HEX).

#### Select FAKECM mode.

- C.2 If Scope Mode set and no error on C.0, C.1, C.2, go to C.3.  
Word 0 from lookahead block.
  - a. Set RF2 to 0--020(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not C3C3 C3C3 C3000004(HEX).

If Scope Mode set, go to C.0.

If C.0, C.1, C.2, error go to C.6.

- C.3 If Scope Mode set and no error on C.3, go to C.4.  
Word 1 from lookahead block.
  - a. Set RF2 to 0--028(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not C3C3 C3C3 C3000005(HEX).

If Scope Mode set, go to C.0.

If C.3 error, go to C.6.

60000285 A

LMT3/P - 453

# Conditions:

- C.4 If Scope Mode set and no error on C.4, go to C.5.  
 Word 2 from lookahead block.  
 a. Set RF2 to 0--030(HEX) (SVA).  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not C3C3 C3C3 C3000006(HEX).

If Scope Mode set, go to C.0.  
 If C.4 error, go to C.8.

- C.5 Word 3 from Lookahead Block.  
 a. Set RF2 to 0--038(HEX) (SVA).  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read RF8, report data; error if not C3C3 C3C3 C3000007(HEX).

If Scope Mode set, go to C.0.

- C.6 CPU Status Word 1.  
 a. Report word 1 if saved; error if reported.

- C.7 CPU Status Word 2.  
 a. Report word 2 if saved; error if reported.

- C.8 First Failure Capture Information.  
 a. Report PFS87 bits 19-23 in bits 59-63; information only.  
 b. Report last PTL written in bits 34-47; information only.  
 c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

## Section 39, Subsection 2

This subsection tests the 27-bit Next Block Address (NXTBA) Incrementer. All gates are tested except for the overflow into address bit 32.

### Hardware Tested:

- NXTBA Bits 32 through 47.
- NXTBA Bits 48 through 58.
- Cache Set 0 Block Used Bit.

# Method:

To test the 27-bit NXTBA Incrementer a sliding zero pattern is used. All SVAs (26) are composed of all 1s and a single 0 bit. The 0 bit is moved through every bit position (SVA bits 32 through 58) sent to the incrementer. This should test all incrementer gates.

All SVAs are designed to result in a Hash Code of zero. This allows the PTA register contents to control the address of the Page Table Descriptor (PTD) in CM. All PTDs are located at CM 1000(HEX).

The Lookahead operation is initiated by issuing a read CM instruction using a SVA containing the current test pattern. A Cache Miss should occur. A second read to the same SVA will result in a hit and a Lookahead to CM address PFA (always zero) plus the Page Offset). This address ranges from 3C(HEX) down to 0.

FAKECM mode is used to read the Lookahead block contents, word zero only, verifying proper addressing.

Cache Set 0 is used exclusively. Therefore, if a Lookahead fails to execute, the Next Block Used Bit (BKUSED) should be suspected along with the NXTBA incrementer.

Condition Two purges the current Cache block and executes the test. Therefore, to repeat the test using the same variables, repeat Condition Two.

## Test Patterns (hexadecimal):

Variables used during the initial CM read.

| Pass | SVA            | PTD                 | RMA    |
|------|----------------|---------------------|--------|
| 1    | FFFF 7FFF FFC0 | FFFF FFFF FFC0 0000 | 000038 |
| 2    | FFFF 7FFF FFA0 | FFFF FFFF FFC0 0000 | 000034 |
| 3    | FFFF 7FFF FF80 | FFFF FFFF FFC0 0000 | 00002C |
| 4    | FFFF 7FFF FEE0 | FFFF FFFF FFC0 0000 | 00001C |
| 5    | FFFF 7FFF FDE0 | FFFF EFFF FF80 0000 | 00003C |
| 6    | FFFF 7FFF FBE0 | FFFF DFFF FF40 0000 | 00003C |
| 7    | FFFF 7FFF F7E0 | FFFF BFFF FEC0 0000 | 00003C |
| 8    | FFFF 7FFF EFE0 | FFFF 7FFF FDC0 0000 | 00003C |
| ↓    | ↓              | ↓                   | ↓      |
| 15   | FFFF 7DFF FFE0 | FFFF FBFF FFC0 0000 | 00003C |
| 16   | FFFF 7BFF FFE0 | FFFF FF7F FFC0 0000 | 00003C |
| 17   | FFFF 77FF FFE0 | FFFF FEFF FFC0 0000 | 00003C |
| 18   | FFFF 6FFF FFE0 | FFFF FDFF FFC0 0000 | 00003C |
| 19   | FFFF 5FFF FFE0 | FFFF FBFF FFC0 0000 | 00003C |
| 1A   | FFFF 3FFF FFE0 | FFFF F7FF FFC0 0000 | 00003C |



Variables used after the increment by the Lookahead operation.

| Pass | SVA            | PTD                 | RMA    |
|------|----------------|---------------------|--------|
| 1    | FFFF 7FFF FFE0 | BFFF FFFF FFC0 0000 | 00003C |
| 2    | FFFF 7FFF FFC0 | BFFF FFFF FFC0 0000 | 000038 |
| 3    | FFFF 7FFF FF80 | BFFF FFFF FFC0 0000 | 000030 |
| 4    | FFFF 7FFF FF00 | BFFF FFFF FFC0 0000 | 000020 |
| 5    | FFFF 7FFF FE00 | BFFF FFFF FFC0 0000 | 000000 |
| 6    | FFFF 7FFF FC00 | BFFF DFFF FF80 0000 | 000000 |
| 7    | FFFF 7FFF F800 | BFFF BFFF FF00 0000 | 000000 |
| 8    | FFFF 7FFF F000 | BFFF 7FFF FE00 0000 | 000000 |
| ↓    | ↓              | ↓                   | ↓      |
| 15   | FFFF 7E00 0000 | BFFF FFC0 0000 0000 | 000000 |
| 16   | FFFF 7C00 0000 | BFFF FF80 0000 0000 | 000000 |
| 17   | FFFF 7800 0000 | BFFF FF00 0000 0000 | 000000 |
| 18   | FFFF 7000 0000 | BFFF FE00 0000 0000 | 000000 |
| 19   | FFFF 6000 0000 | BFFF FC00 0000 0000 | 000000 |
| 1A   | FFFF 4000 0000 | BFFF F800 0000 0000 | 000000 |

CM is initialized by setting a word address pattern into all cells from 0 to FFF(HEX). The expected result is therefore the data from the current RMA.

|        |   |                     |
|--------|---|---------------------|
| CM 0   | = | C3C3 C3C3 C3C3 0000 |
| CM 1   | = | C3C3 C3C3 C3C3 0001 |
| CM 2   | = | C3C3 C3C3 C3C3 0002 |
| ↓      |   | ↓                   |
| CM FFF | = | C3C3 C3C3 C3C3 0FFF |

Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| C      | FAKECM Allocate Cache.                     |
| D      | Read Word from SVA.                        |
| F      | Purge Cache All.                           |
| 1A     | Double Read from two SVAs; II/II Requests. |

Initialization:

- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Cache Set 0 only.
- Set current operand pointer to 0.
- Set pass count to 1.

60000285 A

LMT3/P - 456

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 1A(HEX).  
a. Report pass count; informational.
- C.1 Current SVA.  
a. Set CM 1000(HEX) to first read PTD entry.  
b. Set CM 1001(HEX) to lookahead read PTD entry.  
c. Report current SVA; informational.
- C.2 Execute test, report expected result.  
a. Select FAKECM mode.  
b. Set RF2 to 0000 0000 xxx0(HEX); xxx = current Cache block.  
c. Start processor at micrand C, FAKECM allocate Cache; clear current Cache Tag, wait for CPU halted.  
d. Deselect FAKECM mode.  
e. Read CPU status, format and save if error; error if not 0.  
f. Set RF2 to current SVA.  
g. Set RF3 to current SVA.  
h. Set RF6 to DF--DF(HEX); filler.  
i. Set RF7 to DF--DF(HEX); filler.  
j. Start processor at micrand 1A, double read from two SVAs; II/II requests, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Build and report expected result for current RMA; informational.
- C.3 If Scope Mode set and no error on C.2, C.3, C.4, go to C.5.  
Actual result from first read.  
a. Read RF6, report data; error if not as reported in C.2.
- C.4 Actual Result from second Read.  
a. Read RF7, report data; error if not as reported in C.2.
- If Scope Mode set, go to C.0.  
If C.2, C.3, C.4 error, go to C.8.
- C.5 If Scope Mode set and no C.5 error, go to C.8.  
Actual result from Cache, Word 0, block N.  
a. Select FAKECM mode.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not as reported in C.2.

If Scope Mode set, go to C.0.  
If C.5 error, go to C.8.

60000285 A

LMT3/P - 457

Conditions:

- C.6 Expected result, Word 0, block N+1.  
a. Build and report expected result from current Lookahead RMA; informational.
- C.7 Actual result from Cache, Word 0, block N+1.  
a. Select FAKECM mode.  
b. Set RF2 to current lookahead SVA.  
c. Set RF6 to DF--DF(HEX); filler.  
d. Start processor at micrand D, read word from SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF6, report data; error if not as reported in C.6.

If Scope Mode is set, go to C.0.

- C.8 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

Exit Subsection if 1A(HEX) passes complete; else advance current operand pointer and return to C.0.

Deselect Cache Lookahead Mode.

SECTION 40 - CACHE LOOKAHEAD, PART 2

The Cache Lookahead test is divided into three sections. This section, Part 2, tests the Next Block Used bit in the Cache Tag Status RAM. Each subsection tests one Cache Set.

Subsection 0 - Set 0, Status RAM bit 2 (BKUSED).

Subsection 1 - Set 1, Status RAM bit 2 (BKUSED).

Subsection 2 - Set 2, Status RAM bit 2 (BKUSED).

Subsection 3 - Set 3, Status RAM bit 2 (BKUSED).

Method:

Bit 2 of the Cache Tag Status RAM is tested for data integrity in all 256 locations. This is done by initially clearing the bit using Cache Purge and then setting the bit and testing for the set condition. The following steps are executed, for each of the 256 Cache blocks.

1. Read CM to Block Fill the current Cache block. No Lookahead is performed.
2. Read again at the same address, data is now read from Cache and because bit 2 in the Cache Status RAM is clear, Lookahead is initiated, bit 2 now sets.
3. Check the Lookahead block data in Cache. If Lookahead did not occur, bit 2 probably did not clear during the Cache Purge.
4. Change the data in Cache in the Lookahead block.
5. Repeat step 2. Lookahead should not occur now because bit 2 should be set. Check the Lookahead block data. If Lookahead did occur bit 2 did not set.
6. Advance block address and repeat.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 0.
- Set up Page Table at CM; 1000(HEX), 17 entries.
- Select Unconditional Lookahead.

## Section 40, Subsection 0

This subsection tests the Cache Set 0 Status RAM in all locations for bit two integrity. Bit two controls Cache Lookahead by indicating whether the next Cache Block has already been brought into Cache.

### Hardware Tested:

- Cache Tag Block Used Bit.

### Method:

Refer to section method for details.

All SVAs are arranged to result in a Hash Code of zero. The PTA is set such that all Page Table Descriptors (PTD) start at CM 1000(HEX). The resulting RMAs (PFA from CM always zero) and Page Offset vary from 0 through 3FC(HEX). CM is initialized with word address data in each word. Therefore, the data fetched indicates the RMA generated.

### Test Patterns (hexadecimal):

Variables used during the CM read operations.

| Pass  | SVA            | PTD                 | RMA    |
|-------|----------------|---------------------|--------|
| 1     | 0000 0000 0000 | F000 0000 0000 0000 | 000000 |
| 2     | 0000 0000 0020 | F000 0000 0000 0000 | 000004 |
| 3     | 0000 0000 0040 | F000 0000 0000 0000 | 000008 |
| 4     | 0000 0000 0060 | F000 0000 0000 0000 | 00000C |
| 5     | 0000 0000 0080 | F000 0000 0000 0000 | 000010 |
| 6     | 0000 0000 00A0 | F000 0000 0000 0000 | 000014 |
| 7     | 0000 0000 00C0 | F000 0000 0000 0000 | 000018 |
| 8     | 0000 0000 00E0 | F000 0000 0000 0000 | 00001C |
| 9     | 0000 0000 0100 | F000 0000 0000 0000 | 000020 |
|       |                |                     |        |
| F     | 0000 0000 01E0 | F000 0000 0000 0000 | 00003C |
| 10    | 0000 0000 0200 | F000 0000 0040 0000 | 000040 |
| 11    | 0000 0000 0220 | F000 0000 0040 0000 | 000044 |
| 12    | 0000 0000 0240 | F000 0000 0040 0000 | 000048 |
|       |                |                     |        |
| 100   | 0000 0000 1FE0 | F000 0000 03C0 0000 | 0003FC |
| 100** | 0000 0000 2000 | F000 0000 0400 0000 | 000400 |

\*\*The values of the variables used after the increment by the Lookahead operation are shown in the next pass in the above table. For example in Pass 1 the initial read uses an SVA of 0--0 and the resulting RMA is also zero. The Lookahead SVA and RMA for the same pass are shown in Pass 2; SVA = 0--020(HEX); RMA = 0-04(HEX). That is the reason for the second set of variables at pass 100, the Lookahead variables must be shown.

60000285 A

LMT3/P - 460

CM is initialized as follows.

|        |   |                     |
|--------|---|---------------------|
| CM 0   | = | C3C3 C3C3 C3C3 0000 |
| CM 1   | = | C3C3 C3C3 C3C3 0001 |
| CM 2   | = | C3C3 C3C3 C3C3 0002 |
|        |   |                     |
| CM FFF | = | C3C3 C3C3 C3C3 0FFF |

Write date is formatted as: C3xx yyyy yyyy yyyy(HEX).

xx = 00 (Set 0).  
y-y = Lookahead SVA.

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| F      | Purge Cache All.    |

### Initialization:

- Purge Cache All.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 through FFF(HEX) to C3C3 C3C3 C3C3 0xxx(HEX); xxx = 0 through FFF(HEX).
- Select Cache Set 0 only.
- Block Fill First Cache Block
- Set RF2 to 0--0 (SVA).
- Start processor at micrand D, read word from SVA, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set pass count to 1.

### Conditions:

#### NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 100(HEX)).  
a. Report pass count; informational.

60000285 A

LMT3/P - 461

Conditions:

- C.1 Expected result from lookahead block.  
a. Set RF2 to current SVA.  
b. Build and report expected result for word 0 of lookahead block; informational.
- C.2 If Scope Mode set and no C.2 error, go to C.3.  
Actual result from lookahead block.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Select FAKECM mode.  
e. Set RF6 to DF--DF(HEX); filler.  
f. Set RF2 to current SVA plus 20(HEX); Lookahead SVA.  
g. Start processor to micrand D, read word from SVA, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Read RF6, report data; error if not as reported in C.1.

If Scope Mode set, go to C.0.  
If C.2 error, go to C.5.

- C.3 Change Word 0 of lookahead data, report new data.  
a. Select FAKECM mode.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data.
- C.4 Read again from current SVA, report lookahead data.  
a. Set RF2 to current SVA.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Deselect FAKECM mode.  
d. Start processor at micrand D, read word from SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Select FAKECM mode.  
g. Set RF2 to current SVA plus 20(HEX); lookahead SVA.  
h. Set RF6 to DF--DF(HEX); filler.  
i. Start processor at micrand D, read word from SVA, wait for CPU halted.  
j. Read CPU status, format and save if error; error if not 0.  
k. Read RF6, report data; error if not as reported in C.3.

If Scope Mode set, go to C.0.

- C.5 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.6 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

Conditions:

- C.7 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

Exit Subsection if 100(HEX) passes complete; else advance current operand pointer and return to C.0.

Section 40, Subsection 1

This subsection tests the Cache Set 1 Status RAM in all locations for bit two integrity. Bit two controls Cache Lookahead by indicating whether the next Cache Block has already been brought into Cache.

Hardware Tested:

- Cache Tag Block Used Bit.

Method:

Refer to section method for details.

All SVAs are arranged to result in a Hash Code of zero. The PTA is set such that all Page Table Descriptors (PTD) start at CM 1000(HEX). The resulting RMAs (PFA from CM always zero) and Page Offset) vary from 0 through 3FC HEX). CM is initialized with word address data in each word. Therefore, the data fetched indicates the RMA generated.

Test Patterns (hexadecimal):

Variables used during the CM read operations.

| Pass | SVA            | PTD                 | RMA    |
|------|----------------|---------------------|--------|
| 1    | 0000 0000 0000 | F000 0000 0000 0000 | 000000 |
| 2    | 0000 0000 0020 | F000 0000 0000 0000 | 000004 |
| 3    | 0000 0000 0040 | F000 0000 0000 0000 | 000008 |
| 4    | 0000 0000 0060 | F000 0000 0000 0000 | 00000C |
| 5    | 0000 0000 0080 | F000 0000 0000 0000 | 000010 |
| 6    | 0000 0000 00A0 | F000 0000 0000 0000 | 000014 |
| 7    | 0000 0000 00C0 | F000 0000 0000 0000 | 000018 |
| 8    | 0000 0000 00E0 | F000 0000 0000 0000 | 00001C |
| 9    | 0000 0000 0100 | F000 0000 0000 0000 | 000020 |
|      |                |                     |        |
| F    | 0000 0000 01E0 | F000 0000 0000 0000 | 00003C |

| Pass  | SVA            | PTD                 | RMA    |
|-------|----------------|---------------------|--------|
| 10    | 0000 0000 0200 | F000 0000 0040 0000 | 000040 |
| 11    | 0000 0000 0220 | F000 0000 0040 0000 | 000044 |
| 12    | 0000 0000 0240 | F000 0000 0040 0000 | 000048 |
| ↓     | ↓              | ↓                   | ↓      |
| 100   | 0000 0000 1FE0 | F000 0000 03C0 0000 | 0003FC |
| 100** | 0000 0000 2000 | F000 0000 0400 0000 | 000400 |

\*\*The values of the variables used after the increment by the lookahead operation are shown in the next pass in the above table. For example in Pass 1 the initial read uses an SVA of 0--0 and the resulting RMA is also zero. The Lookahead SVA and RMA for the same pass are shown in Pass 2; SVA = 0--020(HEX); RMA = 0-04(HEX). That is the reason for the second set of variables at pass 100, the Lookahead variables must be shown.

CM is initialized as follows.

|        |   |                     |
|--------|---|---------------------|
| CM 0   | = | C3C3 C3C3 C300 0000 |
| CM 1   | = | C3C3 C3C3 C300 0001 |
| CM 2   | = | C3C3 C3C3 C300 0002 |
| ↓      |   | ↓                   |
| CM FFF | = | C3C3 C3C3 C300 0FFF |

Write data is formatted as: C301 xxxx xxxx xxxx(HEX).

x-x = Lookahead SVA.

Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| F      | Purge Cache All.    |

Initialization:

- Purge Cache All.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 through FFF(HEX) to C3C3 C3C3 C3C3 0xxx(HEX); xxx = 0 through FFF(HEX).
- Select Cache Set 1 only.
- Block Fill First Cache Block.
  - Set RF2 to 0--0 (SVA).
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set pass count to 1.

80000285 A

LMT3/P - 484

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 100(HEX);
  - a. Report pass count; informational.
- C.1 Expected result from lookahead block.
  - a. Set RF2 to current SVA.
  - b. Build and report expected result for word 0 of lookahead block; informational.
- C.2 If Scope Mode set and no C.2 error, go to C.3. Actual result from Cache read.
  - a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Select FAKECM mode.
  - e. Set RF8 to DF--DF(HEX); filler.
  - f. Set RF2 to current SVA plus 20(HEX); Lookahead SVA.
  - g. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Read RF8, report data; error if not as reported in C.1.

If Scope Mode set, go to C.0.  
If error on C.2, go to C.5.

- C.3 Change Word 0 of lookahead data, report new data.
  - a. Select FAKECM mode.
  - b. Set RFA to formatted write data.
  - c. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Report write data.
- C.4 Read again from Current SVA, report lookahead data.
  - a. Set RF2 to current SVA.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Deselect FAKECM mode.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Select FAKECM mode.
  - g. Set RF2 to current SVA plus 20(HEX); Lookahead SVA.
  - h. Set RF8 to DF--DF(HEX); filler.
  - i. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - j. Read CPU status, format and save if error; error if not 0.
  - k. Read RF8, report data; error if not as reported in C.3.

80000285 A

LMT3/P - 485

Conditions:

If Scope Mode set, go to C.0.

- C.5 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.6 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.7 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 100(HEX) passes complete; else advance current operand pointer and return to C.0.

Section 40, Subsection 2

This subsection tests the Cache Set 2 Status RAM in all locations for bit two integrity. Bit two controls Cache Lookahead by indicating whether the next Cache Block has already been brought into Cache. If set 2 not available, skip subsection.

Hardware Tested:

- Cache Tag Block Used Bit.

Method:

Refer to section method for details.

All SVAs are arranged to result in a Hash Code of zero. The PTA is set such that all Page Table Descriptors (PTD) start at CM 1000(HEX). The resulting RMAs (PFA from CM always zero) and Page Offset) vary from 0 through 3FC HEX). CM is initialized with word address data in each word. Therefore, the data fetched indicates the RMA generated.

Test Patterns (hexadecimal):

Variables used during the CM read operations.

| Pass  | SVA            | PTD                 | RMA    |
|-------|----------------|---------------------|--------|
| 1     | 0000 0000 0000 | F000 0000 0000 0000 | 000000 |
| 2     | 0000 0000 0020 | F000 0000 0000 0000 | 000004 |
| 3     | 0000 0000 0040 | F000 0000 0000 0000 | 000008 |
| 4     | 0000 0000 0060 | F000 0000 0000 0000 | 00000C |
| 5     | 0000 0000 0080 | F000 0000 0000 0000 | 000010 |
| 6     | 0000 0000 00A0 | F000 0000 0000 0000 | 000014 |
| 7     | 0000 0000 00C0 | F000 0000 0000 0000 | 000018 |
| 8     | 0000 0000 00E0 | F000 0000 0000 0000 | 00001C |
| 9     | 0000 0000 0100 | F000 0000 0000 0000 | 000020 |
|       |                |                     |        |
| v     | v              | v                   | v      |
| F     | 0000 0000 01E0 | F000 0000 0000 0000 | 00003C |
| 10    | 0000 0000 0200 | F000 0000 0040 0000 | 000040 |
| 11    | 0000 0000 0220 | F000 0000 0040 0000 | 000044 |
| 12    | 0000 0000 0240 | F000 0000 0040 0000 | 000048 |
|       |                |                     |        |
| v     | v              | v                   | v      |
| 100   | 0000 0000 1FE0 | F000 0000 03C0 0000 | 0003FC |
| 100** | 0000 0000 2000 | F000 0000 0400 0000 | 000400 |

\*\*The values of the variables used after the increment by the Lookahead operation are shown in the next pass in the above table. For example in Pass 1 the initial read uses an SVA of 0--0 and the resulting RMA is also zero. The Lookahead SVA and RMA for the same pass are shown in Pass 2; SVA = 0--020 (HEX); RMA = 0-04(HEX). That is the reason for the second set of variables at pass 100, the Lookahead variables must be shown.

CM is initialized as follows:

|        |   |                     |
|--------|---|---------------------|
| CM 0   | = | C3C3 C3C3 C300 0000 |
| CM 1   | = | C3C3 C3C3 C300 0001 |
| CM 2   | = | C3C3 C3C3 C300 0002 |
|        |   |                     |
| v      |   | v                   |
| CM FFF | = | C3C3 C3C3 C300 0FFF |

Write data is formatted as: C302 xxxx xxxx xxxx(HEX).

x--x = Lookahead SVA.

Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| F      | Purge Cache All.    |

#### Initialization:

- Select Cache Set 2 if available; else exit subsection.
- Purge Cache A11.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM; 0 through FFF(HEX) to C3C3 C3C3 C3C3 0xxx(HEX); xxx = 0 through FFF(HEX).
- Block Fill First Cache Block.
  - Set RF2 to 0--0 (SVA).
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).

#### Conditions:

##### NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 100(HEX)).  
a. Report pass count; informational.
- C.1 Expected result from lookahead block.  
a. Set RF2 to current SVA.  
b. Build and report expected result for word 0 of lookahead block; informational.
- C.2 If Scope Mode set, and no C.2 error, go to C.3.  
Actual result from lookahead block.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Select FAKECM mode.  
e. Set RF8 to DF--DF(HEX); filler.  
f. Set RF2 to current SVA plus 20(HEX); Lookahead SVA.  
g. Start processor at micrand D, read word from SVA, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Read RF8, report data; error if not as reported in C.1.

If Scope Mode is set, go to C.0.  
If error on C.2, go to C.5.

#### Conditions:

- C.3 Change Word 0 of lookahead data, report new data.  
a. Select FAKCM mode.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report write data.
- C.4 Read again from current SVA, report lookahead data.  
a. Set RF2 to current SVA.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Deselect FAKECM mode.  
d. Start processor at micrand D, read word from SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Select FAKECM mode.  
g. Set RF2 to current SVA plus 20(HEX); Lookahead SVA.  
h. Set RF8 to DF--DF(HEX); filler.  
i. Start processor at micrand D, read word from SVA, wait for CPU halted.  
j. Read CPU status, format and save if error; error if not 0.  
k. Read RF8, Report data; error if not as reported in C.3.

If Scope Mode is set, go to C.0.

- C.5 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.6 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.7 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

Exit Subsection if 100(HEX) passes complete; else advance current operand pointer and return to C.0.

#### Section 40, Subsection 3

This subsection tests the Cache Set 3 Status RAM in all locations for bit two integrity. Bit two controls Cache Lookahead by indicating whether the next Cache Block has already been brought into Cache. If set 3 not available, skip subsection.

#### Hardware Tested:

- Cache Tag Block Used Bit.

# Method:

Refer to section method for details.

All SVAs are arranged to result in a Hash Code of zero. The PTA is set such that all Page Table Descriptors (PTD) start at CM 1000(HEX). The resulting RMAs (PFA from CM always zero) and Page Offset) vary from 0 through 3FC(HEX). CM is initialized with word address data in each word. Therefore, the data fetched indicates the RMA generated.

## Test Patterns (hexadecimal):

Variables used during the CM read operations.

| Pass  | SVA            | PTD                 | RMA    |
|-------|----------------|---------------------|--------|
| 1     | 0000 0000 0000 | F000 0000 0000 0000 | 000000 |
| 2     | 0000 0000 0020 | F000 0000 0000 0000 | 000004 |
| 3     | 0000 0000 0040 | F000 0000 0000 0000 | 000008 |
| 4     | 0000 0000 0080 | F000 0000 0000 0000 | 00000C |
| 5     | 0000 0000 0080 | F000 0000 0000 0000 | 000010 |
| 6     | 0000 0000 00A0 | F000 0000 0000 0000 | 000014 |
| 7     | 0000 0000 00C0 | F000 0000 0000 0000 | 000018 |
| 8     | 0000 0000 00E0 | F000 0000 0000 0000 | 00001C |
| 9     | 0000 0000 0100 | F000 0000 0000 0000 | 000020 |
| ↓     | ↓              | ↓                   | ↓      |
| V     | V              | V                   | V      |
| F     | 0000 0000 01E0 | F000 0000 0000 0000 | 00003C |
| 10    | 0000 0000 0200 | F000 0000 0040 0000 | 000040 |
| 11    | 0000 0000 0220 | F000 0000 0040 0000 | 000044 |
| 12    | 0000 0000 0240 | F000 0000 0040 0000 | 000048 |
| ↓     | ↓              | ↓                   | ↓      |
| V     | V              | V                   | V      |
| 100   | 0000 0000 1FE0 | F000 0000 03C0 0000 | 0003FC |
| 100** | 0000 0000 2000 | F000 0000 0400 0000 | 000400 |

\*\*The values of the variables used after the increment by the Lookahead operation are shown in the next pass in the above table. For example, in Pass 1 the initial read uses an SVA of 0--0 and the resulting RMA is also zero. The Lookahead SVA and RMA for the same pass are shown in Pass 2; SVA = 0--020(HEX); RMA = 0-04(HEX). That is the reason for the second set of variables at pass 100, the Lookahead variables must be shown.

CM is initialized as follows.

|        |   |                     |
|--------|---|---------------------|
| CM 0   | = | C3C3 C3C3 C300 0000 |
| CM 1   | = | C3C3 C3C3 C300 0001 |
| CM 2   | = | C3C3 C3C3 C300 0002 |
| ↓      |   | ↓                   |
| CM FFF | = | C3C3 C3C3 C300 0FFF |

60000285 A

LMT3/P - 470

Write data is formatted as: C303 xxxx xxxx xxxx(HEX).

x--x = Lookahead SVA.

Micrands Used:

Number Description

D Read Word from SVA.  
E Write Word to SVA.  
F Purge Cache All.

Initialization:

- Select Cache Set 3 if available; else exit subsection.
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM; 0 through FFF(HEX) to C3C3 C3C3 C3C3 0xxx(HEX); xxx = 0 through FFF(HEX).
- Block Fill First Cache Block.
  - Set RF2 to 0--0 (SVA).
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set pass count to 1.

Conditions:

## NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 100(HEX).  
a. Report pass count; informational.
- C.1 Expected result from lookahead block.  
a. Set RF2 to current SVA.  
b. Build and report expected result for word 0 of lookahead block; informational.

60000285 A

LMT3/P - 471



Conditions:

- C.2 If Scope Mode is set and no C.2 error, go to C.3.  
Actual result from lookahead block.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU Status, format and save if error; error if not 0.  
d. Select FAKECM mode.  
e. Set RF8 to DF--DF(HEX); filler.  
f. Set RF2 to current SVA plus 20(HEX); Lookahead SVA.  
g. Start processor at micrand D, read word from SVA, wait for CPU halted.  
h. Read CPU Status, format and save if error; error if not 0.  
i. Read RF8, report data; error if not as reported in C.1.

If Scope Mode is set, go to C.0.

If error on C.2, go to C.5.

- C.3 Change Word 0 of lookahead data, report new data.  
a. Select FAKECM mode.  
b. Set RFA to formatted write data.  
c. Start processor at micrand E, write word to SVA, wait for CPU halted.  
d. Read CPU Status, format and save if error; error if not 0.  
e. Report write data.
- C.4 Read again from current SVA, report lookahead data.  
a. Set RF2 to current SVA.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Deselect FAKECM mode.  
d. Start processor at micrand D, read word from SVA, wait for CPU halted.  
e. Read CPU Status, format and save if error; error if not zero.  
f. Select FAKECM mode.  
g. Set RF2 to current SVA plus 20(HEX); lookahead SVA.  
h. Set RF8 to DF--DF(HEX); filler.  
i. Start processor at micrand D, read word from SVA, wait for CPU halted.  
j. Read CPU Status, format and save if error; error if not 0.  
k. Read RF8, report data; error if not as reported in C.3.

If Scope Mode set, go to C.0.

- C.5 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.6 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.7 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

Exit Subsection if 100(HEX) passes complete; else advance current operand pointer and return to C.0.

Deselect Lookahead mode.

## SECTION 41 - CACHE LOOKAHEAD, PART 3

The Cache Lookahead test is divided into three sections. This section, Part 3, tests some of the end case logic such as inhibit page fault. It also tests the Conditional, Unconditional selection.

Testing of the PTCT Counter is included in the Lookahead tests because Lookahead must be used to test it.

Subsection 0 - Inhibit Page Fault After Lookahead.

Subsection 1 - Unconditional/Conditional Lookahead Select.

Subsection 2 - Memory Requests Outstanding, Page Table Counter (PTCT).

Method:

Refer to subsection methods.

Initialization:

- Select All Map Sets.
- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 78(HEX).
- Select Unconditional Lookahead.

### Section 41, Subsection 0

This subsection tests the inhibit of the Page Fault signal if the Lookahead address is not in the Page Map or Page Table.

Hardware Tested:

- Lookahead Control.
- Next Block Address (NXTBA) bit 32.

Method:

Conditions 0 through 7 test the effect of Lookahead on the Page Fault logic. If a Lookahead operation is initiated and the resultant address from the NXTBA Incrementer is not in CM no page match) the Page Fault Signal, which would normally result, is inhibited and the Lookahead is terminated gracefully. This situation is tested using both Instruction Issue Requests and Instruction Fetch Requests.

Test Patterns (hexadecimal):

| Condition | SVA            | PTD                 | RMA    |
|-----------|----------------|---------------------|--------|
| 0/1       | 0000 0000 01E0 | B000 0000 0000 0000 | 00003C |
| 4/5       | 0000 0000 07E0 | B000 0000 0000 0000 | 0000FC |

CM 0 through FFF(HEX) is initialized as follows:

|        |   |                     |
|--------|---|---------------------|
| CM 0   | = | C3C3 C3C3 C3C3 0000 |
| CM 1   | = | C3C3 C3C3 C3C3 0001 |
| CM 2   | = | C3C3 C3C3 C3C3 0002 |
|        |   | ↓                   |
| CM FFF | = | C3C3 C3C3 C3C3 0FFF |

Micrands Used:

| Number | Description                |
|--------|----------------------------|
| C      | FAKECM Allocate Cache.     |
| D      | Read Word From SVA.        |
| F      | Purge Cache All.           |
| 10     | Set P Descriptor register. |
| 13     | Set P register; start IF.  |
| E      | Write word to SVA.         |

Initialization:

- Set up Page Table at CM 1000(HEX), three different entries.
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Cache Set 0 only.
- Set P Descriptor register to 0--0.
  - Set RF11 to 0--0; Segment zero.
  - Set CM 0 to 0--0; P Descriptor Value.
  - Start processor at micrand 10, Set P Descriptor, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 0 through FFF(HEX) to C3C3 C3C3 C3C3 0xxx(HEX); xxx = 0 through FFF(HEX).

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.0, C.1, C.2, C.3 error, go to C.4.  
Fault test, fill block F(HEX) and report data read.  
a. Set RF2 to 0--01E0(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3C3 C3C3 C3C3 003C(HEX).
- C.1 Execute II fault test, report data read.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read RF8, report data; error if not as reported in C.4.
- C.2 CPU Status Word 1 from II fault test.  
a. Read CPU status, format and report word 1; error if not 0.
- C.3 CPU Status Word 2 from II fault test.  
a. Report word 2; error if not 0.
- If Scope Mode set, go to C.0.  
If error on C.0, C.1, C.2, C.3, go to C.8.
- C.4 Fill block 3F(HEX), report data read.  
a. Set RF2 to 0--07E0(HEX) (SVA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3C3 C3C3 C3C3 00FC(HEX).
- C.5 Execute IF fault test, report data fetched.  
a. Set RF10 to 0--07E0 (PVA).  
b. Set RF14 to DF--DF(HEX); filler.  
c. Start processor at micrand 13, set P register; start IF, wait for CPU halted.  
d. Read RF14, report data; error if not A5--A5C3C3(HEX).
- C.6 CPU Status Word 1 from IF Fault Test.  
a. Read CPU status, format and report word 1; error if not 0.
- C.7 CPU Status Word 2 from IF Fault Test.  
a. Report word 2; error if not 0.

If Scope Mode set, go to C.0.

60000285 A

LMT3/P - 476

Conditions:

- C.8 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.9 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.10 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect Lookahead.

Section 41, Subsection 1

This subsection ensures that Unconditional Lookahead is not aborted by a CPU request and that Conditional Lookahead is aborted when a CPU request conflicts with its request.

Hardware Tested:

- Lookahead Control.

Method:

To test Conditional versus Unconditional Lookahead a micrand sequence consisting of two II requests is issued twice. Once with Conditional Lookahead enabled and once with Unconditional Lookahead enabled.

Conditional Lookahead is tested first. The test sequence is issued and the Lookahead started by the first II Request should be aborted when the second II Request is received. To test whether Lookahead was completed or not, the Lookahead block in Cache is filled with a known pattern prior to running the test sequence and read after the sequence is complete. If the data is changed, Lookahead is assumed to have completed, and an error is logged.

Unconditional Lookahead is tested in a like manner, only the data in the Lookahead block is expected to be changed. All three operations should complete - both IIs, and the Lookahead.

Test Patterns:

Refer to Conditions.

60000285 A

LMT3/P - 477

# Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| C      | FAKECM Allocate Cache                      |
| D      | Read Word from SVA.                        |
| E      | Write Word to SVA.                         |
| F      | Purge Cache All.                           |
| 13     | Set P register; start IF.                  |
| 1C     | Double Read from two SVAs; II/IF Requests. |

## Initialization:

- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Cache Set 0.
- Set CM, (1000(HEX)), to B000 0000 0000 0000(HEX); PTD entry.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode is set and no C.0, C.1, C.2, C.3, C.4 error, go to C.5. Fill lookahead block; block 1 in Cache, report data.
- a. Set RF2 to 0--020020(HEX) (SVA).
  - b. Select FAKECM mode.
  - c. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not zero.
  - e. Set RFA to E7E7 0000 E7E7 0004(HEX); write data.
  - f. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - g. Read CPU status, format and save if error; error if not 0.
  - h. Deselect FAKECM mode.
  - i. Report write data; informational.
- C.1 Fill Cache block 0, report read data.
- a. Set RF2 to 0--0 (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not C3C3 C3C3 C300 0000(HEX).

60000285 A

LMT3/P - 478

## Conditions:

- C.2 Execute conditional lookahead, report II result.
- a. Select conditional lookahead.
  - b. Set RF3 to 0--0100(HEX); SVA for second II.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Set RF7 to DF--DF(HEX); filler.
  - e. Start processor at micrand 1A, double read from 2 SVAs; II/II Requests, wait for CPU halted.
  - f. Read CPU status, format and save if error; error if not 0.
  - g. Read RF6, report data; error if not C3C3 C3C3 C300 0000(HEX).
- C.3 Report second II result.
- a. Read RF7, report data; error if not C3--C3000020(HEX).
- C.4 Report Word 0 from Cache Lookahead block.
- a. Set RF2 to 0--020020(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Select FAKECM mode.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Deselect FAKECM mode.
  - g. Read RF6, report data; error if not E7E7 0000 E7E7 0004(HEX).
- If Scope Mode is set, go to C.0.  
If error on C.0, C.1, C.2, C.3, C.4, go to C.10.
- C.5 Fill lookahead block; block 1 in Cache.
- a. Start processor at micrand F, purge Cache all, wait for CPU halted.
  - b. Read CPU status, format and save if error; error if not 0.
  - c. Select FAKECM mode.
  - d. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Set RFA to E7E7 0000 E7E7 0004(HEX); write data.
  - g. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Deselect FAKECM mode.
  - j. Report write data; informational.
- C.6 Fill Cache block 0, report read data.
- a. Set RF2 to 0--0 (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF6, report data; error if not C3C3 C3C3 C300 0000(HEX).

60000285 A

LMT3/P - 479

# Conditions:

- C.7 Execute unconditional lookahead, report II result.
  - a. Select unconditional lookahead.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Set RF7 to DF--DF(HEX); filler.
  - d. Start processor at micrand 1A, double read from two SVAs; II/II requests, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF6, report data; error if not C3C3 C3C3 C300 0000(HEX).
- C.8 Report second result.
  - a. Read RF7, report data; error if not C3--C3000020(HEX)
- C.9 Report Word 0 from Cache lookahead block.
  - a. Set RF2 to 0--020(HEX) (SVA).
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Select FAKECM mode.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Deselect FAKECM mode.
  - g. Read RF6, report data; error if not C3C3 C3C3 C300 0004(HEX).

If Scope Mode is set, go to C.0.

- C.10 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.11 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Deselect Lookahead.

## Section 41, Subsection 2

This subsection tests the Page Table Counter (PTCT) to ensure it increments away from zero and decrements back to zero. The actual count cannot be tested. This subsection also tests the circuitry used to block lookahead if lookahead is currently in progress.

### Hardware Tested:

- PTCT Incrementer.

60000285 A

LMT3/P - 480

# Method:

The objective of this subsection is to test the No Memory Requests Outstanding (NMRO) status. This is accomplished in three steps.

1. Test the NMRO status with no memory micrands active to ensure PTCT = 0.
2. Test the NMRO status with CM requests outstanding to ensure PTCT can be nonzero.
3. Use multiple requests and Lookahead to advance the PTCT incrementer as far as possible (about 8), wait for CPU halted and check it for zero.

Step 1 simply issues a micrand to sense the NMRO status. If the NMRO status is true (as it should be) the micrand sets RF6 to E7--E7(HEX); else it is left as originally filled.

Step 2 issues a double Cache Read to the same SVA causing a Lookahead operation. NMRO status is again sensed. This time RF6 should be left as set from the Cache Read, not overwritten with E7--E7(HEX).

Step 3 is tested by issuing a micrand sequence to activate as many CM requests as possible. The first request (II) results in a Cache Hit which initiates a Lookahead. The second request (IF) starts a Block Fill as does the third request (II). When CPU halted is sensed the NMRO status is taken. No requests to CM should be outstanding.

Condition 6 ensures that a Lookahead request is blocked when a previous Lookahead request is still selected. The procedure is:

1. Purge All Cache.
2. Issue two word load operations to fill two different Cache blocks.
3. Reissue the same two word load operations. This time both result in Cache hits and both are eligible for Lookahead. The second Lookahead request should be blocked.

Condition 9 reports the first Cache word from the second requests presumed Lookahead block. A Cache miss (zeros) is expected.

CM refresh is synchronized to ensure consistent results.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description                                     |
|--------|-------------------------------------------------|
| F      | Purge Cache All.                                |
| 1A     | Double Read from two SVAs; II/II Requests.      |
| 25     | Sense NMRO Status.                              |
| 26     | Triple Read from Three SVAs; II/IF/II Requests. |

60000285 A

LMT3/P - 481

#### Initialization:

- Purge Cache All.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 1000(HEX) to B000 0000 0000 0000(HEX) (PTD Entry).
- Select Unconditional Lookahead.

#### Conditions:

##### NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.0, C.1, C.2, C.3, C.4, C.5 error, go to C.6. Test NMRO true, report RF8.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 25, sense NMRO status, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not E7--E7(HEX).
- C.1 Test NMRO false, report RF8.
- a. Set RF2 to 0--0; SVA, first read.
  - b. Set RF3 to 0--0; SVA, second read.
  - c. Set RF8 to DF--DF(HEX); filler.
  - d. Set RF7 to DF--DF(HEX); filler.
  - e. Start processor at micrand 1A, double read from two SVAs; II/II request.
  - f. Start processor at micrand 25, sense NMRO status, wait for CPU halted.
  - g. Read CPU status, format and save if error; error if not 0.
  - h. Read RF8, report data; error if not C3C3 C3C3 C3C3 0000(HEX).
- C.2 Issue II/IF/II sequence, report first II result.
- a. Set RF2 to 0--020; SVA for first II - block one.
  - b. Set RF10 to 0--080; PVA for IF - block three.
  - c. Set RF3 to 0--0A0; SVA for second II - block five.
  - d. Set RF8 to DF--DF(HEX); filler.
  - e. Set RF7 to DF--DF(HEX); filler.
  - f. Set RF14 to DF--DF(HEX); filler.
  - g. Start processor at micrand 28, triple read from three SVAs; II/IF/II requests.
  - h. Read CPU status, format and save if error; error if not 0.
  - i. Read RF8, report data, error if not C3C3 C3C3 C3C3 0004(HEX).
- C.3 Report IF result.
- a. Read RF14, report data; error if not A5A5 A5A5 A5A5 C3C3(HEX).

#### Conditions:

- C.4 Report second II result.
- a. Read RF7, report data; error if not C3C3 C3C3 C3C3 0014(HEX).
- C.5 Test NMRO true, report RF8.
- a. Set RF8 to DF--DF(HEX); filler.
  - b. Start processor at micrand 25, sense NMRO status, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read RF8, report data; error if not E7--E7(HEX).
- If Scope Mode is set, go to C.0.  
If error on C.0, C.1, C.2, C.3, C.4, C.5, go to C.10.
- C.6 Test block lookahead when lookahead active.
- a. Start processor at micrand F, purge all Cache, wait for CPU halted.
  - b. Read CPU Status, format and save if error; error if not 0.
  - c. Set RF2 to 0--0; first SVA.
  - d. Set RF3 to 0--040(HEX); second SVA.
  - e. Start processor at micrand 1A, double word load; block fill two Cache blocks, wait for CPU halted.
  - f. Read CPU Status, format and save if error; error if not 0.
  - g. Set RF8 to DF--DF(HEX); filler.
  - h. Set RF7 to DF--DF(HEX); filler.
  - i. Start processor at micrand 1A, double word load; Cache hits request lookahead, wait for CPU halted.
  - j. Read CPU Status, format and save if error; error if not 0.
  - k. Read RF8, report data; error if not C3--C3000000(HEX).
- C.7 Report Cache word from first lookahead.
- a. Select FAKECM mode.
  - b. Set RF2 to 0--020(HEX); SVA, block 1.
  - c. Set RF8 to DF--DF(HEX); filler.
  - d. Start processor at micrand D, load word; read from Cache block 1, word 0, wait for CPU halted.
  - e. Read CPU Status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000004(HEX).
- C.8 Report result from second read.
- a. Read RF7, report data; error if not C3--C3000008(HEX).
- C.9 Report Cache word from second lookahead area.
- a. Set RF2 to 0--080(HEX); SVA, block 3.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, load word; read Cache wait for CPU halted.
  - d. Read CPU Status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not 0.
- If Scope Mode is set, go to C.0.
- C.10 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.

Conditions:

- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect Unconditional Lookahead.

Deselect FAKECM Mode.

SECTION 42 - PAGE TABLE UPDATE

This section tests the Page Table Update mechanism, ensuring the Used bit sets when a Page Table Hit occurs, and the Modify bit sets when a write is performed to an address located on the matched page. This section also tests the decrement feature within the RMA Incrementer.

Subsection 0 - Update Used Bit in Page Table.

Subsection 1 - RMA Decrement for Update Address.

Subsection 2 - Update Modify During a Map Miss Write.

Subsection 3 - Update Modify in Map From Page Table.

Method:

Refer to subsection methods.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Select All Map Sets.

Section 42, Subsection 0

This subsection tests the Page Table Update logic that sets the Used bit, bit 2, in the Page Table Descriptor after a Page Table Hit is sensed.

Hardware Tested:

- Update Control.
- Address/Data Output mux.

Method:

The objective of this subsection is to test the basic Page Table Update operation. This is done by setting up a Page Table in which all descriptors have only the Valid (bit 0) and Continue (bit 1) bits set. A read operation is issued and a Page Table Hit occurs. Byte 0 of the Descriptor is rewritten setting the Used bit (bit 2). The Page Map is written and the requested word is read to the Register File.

This basic sequence is repeated three times; an odd descriptor is used, an even descriptor is used (first use of RMA decrement), and the Test function is used.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| 12     | Purge Map All.      |
| 15     | Test Operation.     |

Initialization:

- Deselect all Cache Sets.
- Set CM 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Set up Page Table at CM 1000(HEX).  
CM 1000(HEX) = C000 0000 0000 0000(HEX)  
CM 1001(HEX) = C010 0000 0000 0000(HEX)  
CM 1002(HEX) = C020 0000 0000 0000(HEX)  

↓                      ↓  
CM 101F(HEX) = C1F0 0000 0000 0000(HEX)
- Purge Cache All.
- Start processor at micrand F, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FF(HEX)).
- Purge Map All.
- Start processor at micrand 12, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FF(HEX)).

Conditions:

NOTE

Disregard skipped conditions on error display.

60000285 A

LMT3/P - 486

Conditions:

- C.0 If Scope Mode set and no C.4, C.5 error, go to C.6.  
Report odd descriptor before update.  
a. Read CM; (1001(HEX)), report data; informational.
- C.1 Report odd descriptor after update.  
a. Set RF2 to 0100 0000 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM; (1001(HEX)), report data; error if not E010 0000 0000 0000(HEX).
- C.2 Report read data from C.1.  
a. Read RF8, report data; error if not C3C3 C3C3 C300 0000(HEX).
- If Scope Mode set, go to C.0.  
If Scope Mode error on C.1, C.2, go to C.9.
- C.3 If Scope Mode set and no C.4, C.5 error, go to C.6.  
Report even descriptor before update.  
a. Read CM 1000(HEX), report data; informational.
- C.4 Report even descriptor after update.  
a. Set RF2 to 0000 0000 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 1000(HEX), report data; error if not E000 0000 0000 0000(HEX).
- C.5 Report read data from C.4.  
a. Read RF8, report data; error if not C3C3 C3C3 C300 0000(HEX).
- If Scope Mode set, go to C.0.  
If error on C.4, C.5 go to C.9.
- C.6 Test function, report odd descriptor before update.  
a. Read CM; (1003(HEX)), report data; informational.
- C.7 Report odd descriptor after update.  
a. Set RF2 to 0300 0000 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand 15, Test, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM; (1003(HEX)), report data; error if not E030 0000 0000 0000(HEX).
- C.8 Report read data from C.7.  
a. Read RF8, report data; error if not 0.

If Scope Mode set, go to C.0.

60000285 A

LMT3/P - 487



# Conditions:

- C.9 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

## Section 42, Subsection 1

Complete testing of the Decrement RMA logic contained in the RMA Incrementer.

### Hardware Tested:

- RMA Increment/Decrement.

### Method:

The RMA Incrementer was tested using incrementing addresses in a previous section. This subsection tests the incrementer to ensure it correctly decrements when requested to do so. The lower three RMA bits are affected by the decrement signal. Therefore, only five operands are necessary to test all decrement cases.

Page Table Hits are set up to occur on Page Table Descriptor entries 0 through 4. Decrement should only occur when a hit occurs on the even descriptors (entries 0, 2, and 4).

The Used bit is checked both before and after the Update. If an error occurs the Used bit in the wrong PTD entry will likely be set.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| F      | Purge Cache All.    |
| 12     | Purge Map All.      |

60000285 A

LMT3/P - 488

# Initialization:

- Select all Available Cache Sets.
- Set up Page Table at CM 1000(HEX).  
 CM 1000(HEX) = C000 0000 0000 0000(HEX).  
 CM 1001(HEX) = C010 0000 0000 0000(HEX).  
 CM 1002(HEX) = C020 0000 0000 0000(HEX).  
 ↓ ↓  
 CM 101F(HEX) = C1F0 0000 0000 0000(HEX).  
 - Purge Cache All.  
 - Start processor at micrand F, wait for CPU halted.  
 - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).  
 - Purge Map All.  
 - Start processor at micrand 12, wait for CPU halted.  
 - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).

### Conditions:

#### NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.1 error, go to C.2.  
 Report descriptor 0 before update.  
 a. Read CM 1000(HEX), report data; informational.
- C.1 Report descriptor 0 after update.  
 a. Set RF2 to 0--0 (SVA).  
 b. Set RF8 to DF--DF(HEX); filler.  
 c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
 d. Read CPU status, format and save if error; error if not 0.  
 e. Read CM 1000(HEX), report data; error if not E000 0000 0000(HEX).

If Scope Mode set, go to C.0.  
 If error on C.1, go to C.10.

- C.2 If Scope Mode set and no C.3 error, go to C.4.  
 Report descriptor 1 before update.  
 a. Read CM; (1001(HEX)), report data; informational.

60000285 A

LMT3/P - 489

Conditions:

- C.3 Report descriptor 1 after update.  
a. Set RF2 to 0100 0000 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM; (1001(HEX)), report data; error if not E010 0000 0000(HEX).

If Scope Mode set, go to C.0.  
If error on C.3, go to C.10.

- C.4 If Scope Mode set and no C.5 error, go to C.6.  
Report descriptor 2 before update.  
a. Read CM; (1002(HEX)), report data; informational.

- C.5 Report descriptor 2 after update.  
a. Set RF2 to 0200 0000 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM; (1002(HEX)), report data; error if not E020 0000 0000(HEX).

If Scope Mode set, go to C.0.  
If error on C.5, go to C.10.

- C.6 If Scope Mode set and no C.7 error, go to C.8.  
Report descriptor 3 before update.  
a. Read CM; (1003(HEX)), report data; informational.

- C.7 Report descriptor 3 after update.  
a. Set RF2 to 0300 0000 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM; (1003(HEX)), report data; error if not E030 0000 0000(HEX).

If Scope Mode set, go to C.0.  
If error on C.7, go to C.10.

- C.8 Report descriptor 4 before update.  
a. Read CM; (1004(HEX)), report data; informational.

Conditions:

- C.9 Report descriptor 4 after update.  
a. Set RF2 to 0400 0000 0000(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM; (1004(HEX)), report data; error if not E040 0000 0000(HEX).

If Scope Mode is set, go to C.0.

- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Section 42, Subsection 2

This subsection tests Page Table Update logic when both the Table and the Map are updated during a write operation. This subsection tests all three Page Map Sets.

Hardware Tested:

- Update Control.
- Address/Data mux.
- Modify Logic.

Method:

This subsection tests the logic that sets the Modify bit in the Page Table and the Page Map when a Map Miss occurs during a write operation.

Two write micrands, using the same address, are required. The first to set the Modify bit in the Page Table Descriptor (PTD) and in the Page Map. The second is used to test the Page Map proving the bit did set. This is done by clearing the Modify bit in the PTD after the first write, and checking that it remains clear after the second write. If the bit did not set in the Map, the second write would result in a Page Table Update resetting the Modify bit.

All three sets are tested using three conditions per set. The set to test is selected and the PTD entered into CM. The PTD has only the valid bit set, this value is reported. A write is executed and the Used and Modify bits should set. The PTD is again reported. The Modify bit is cleared and a second write executed. The Modify bit should remain clear; the PTD is again reported.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write Word to SVA. |
| 12     | Purge Map All.     |

#### Initialization:

- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1 (EF(HEX)).

#### Conditions:

##### NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.2 error, go to C.3.  
Report initial PTD entry, Set 0.
- a. Set CM; (1000(HEX) to 8000 0000 0000 0000(HEX); PTD entry.
  - b. Select Map Set 0.
  - c. Report PTD entry; informational.
- C.1 Set Modify bit in PTD and Map, Set 0.
- a. Set RF2 to 0--0 (SVA).
  - b. Set RFA to E7--E7(HEX); write data.
  - c. Set CM 0 = C3--C3000000(HEX).
  - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read CM 1000(HEX), report data; error if not B000 0000 0000 0000(HEX).

60000285 A

LMT3/P - 492

#### Conditions:

- C.2 Test Modify bit on Map, Set 0.
- a. Set CM 0 = C3--C3000000(HEX).
  - b. Set CM 1000(HEX) to A000 0000 0000 0000(HEX); clear Modify.
  - c. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read CM 1000(HEX), report data; error if not A000 0000 0000 0000(HEX); Modify stays clear.

If Scope Mode set, go to C.0.

If error on C.1, C.2, go to C.9.

- C.3 If Scope Mode set and no C.4, C.5 error, go to C.6.  
Report initial PTD entry, Set 1.
- a. Set CM 1000(HEX) to 8000 0000 0000 0000(HEX); PTD entry.
  - b. Select Map Set 1.
  - c. Report PTD entry; informational.
- C.4 Set Modify bit in PTD and Map, Set 1.
- a. Set CM=0 = C3-C3000000(HEX).
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 1000(HEX), report data; error if not B000 0000 0000 0000(HEX).
- C.5 Test Modify bit on Map, Set 1.
- a. Set CM 0 = C3--C3000000(HEX).
  - b. Set CM 1000(HEX) to A000 0000 0000 0000(HEX); clear Modify.
  - c. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read CM 1000(HEX), report data; error if not A000 0000 0000 0000(HEX); Modify stays clear.

If Scope Mode set, go to C.0.

If error on C.4, C.5, go to C.9.

- C.6 If Scope Mode set and no C.7, C.8 error, go to C.9.  
Report initial PTD entry, Set 2.
- a. Set CM 1000(HEX) to 8000 0000 0000 0000(HEX); PTD entry.
  - b. Select Map Set 2.
  - c. Report PTD entry; informational.
- C.7 Set Modify in PTD and Map, Set 2.
- a. Set CM 0 = C3--C3000000(HEX).
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 1000(HEX), report data; error if not B000 0000 0000 0000(HEX).

60000285 A

LMT3/P - 493

# Conditions:

- C.8 Test Modify bit on Map, Set 2.
- Set CM 0 = C3--C3000000(HEX)
  - Set CM 1000(HEX) to A000 0000 0000 0000(HEX); clear Modify.
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 1000(HEX), report data; error if not A000 0000 0000 0000(HEX); Modify stays clear.

If Scope Mode set, go to C.0.

- C.9 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

## Section 42, Subsection 3

This subsection tests the Modify bit path (CM (3)), from the Page Table Descriptor (PTD) in CM to the Page Map. It also tests the Modify bit recycle path from Map RAM back to the Map RAM.

### Hardware Tested:

- (CM (3)).
- Page Map Modify Logic.
- Write Map After Map Hit.

### Method:

This subsection tests two means of setting the Modify bit in the Page Map. The first method is when a Map Miss occurs during a read operation and the Modify bit is set in the PTD. The second method is when the Modify bit is set in the Map and a read operation is issued. The Modify bit must be recycled; i.e., read out and rewritten to save the Modify status.

All three sets are tested using three conditions per set. The set to test is selected and a PTD containing the Valid, Used and Modify bits is set into CM (1000(HEX)), and reported. A read operation is issued and the Modify bit should be written into the Map. A second read operation is issued which should recycle the Modify status. To ensure the Modify bit is still set in the Map, the Modify bit in the PTD is cleared and reported, and a write operation is issued. The Modify bit in the PTD should remain clear. This final PTD is also reported.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| 12     | Purge Map All.      |

### Initialization:

- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Deselect all Cache Sets.

### Conditions:

#### NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.1, C.2 error, go to C.3.
- Report initial PTD entry, Set 0.
- Select Map Set 0.
  - Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry.
  - Report PTD entry; informational.

Conditions:

- C.1 Execute two reads, clear Modify bit and report PTD, Set 0.
- Set RF2 to 0--0 (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set CM 1000(HEX) to A000 0000 0000 0000(HEX); clear Modify in PTD.
  - Report PTD entry; informational.

- C.2 Test Modify bit, report PTD, Set 0.
- Set CM 0 to C3--C3000000(HEX).
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 1000(HEX), report data; error if not A000 0000 0000 0000(HEX).

If Scope Mode set, go to C.0.  
If error on C.1, C.2, go to C.9.

- C.3 If Scope Mode set and no C.4, C.5, go to C.6.  
Report initial PTD entry, Set 1.
- Select Map Set 1.
  - Set CM 1000(HEX) to 8000 0000 0000 0000(HEX); PTD entry.
  - Report PTD entry; informational.

- C.4 Execute two reads, clear Modify bit and report PTD, Set 1.
- Set RF2 to 0--0 (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set CM 1000(HEX) to A000 0000 0000 0000(HEX); clear Modify in PTD.
  - Report PTD entry; informational.

- C.5 Test Modify bit, report PTD, Set 1.
- Set CM 0 to C3--C3000000(HEX).
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 1000(HEX), report data; error if not A000 0000 0000 0000(HEX).

Conditions:

If Scope Mode set, go to C.0.  
If error on C.4, C.5, go to C.9.

- C.6 If Scope Mode set and no C.7, C.8 error, go to C.9.  
Report initial PTD entry, Set 2.
- Select Map Set 2.
  - Set CM 1000(HEX) to 8000 0000 0000 0000(HEX); PTD entry.
  - Report PTD entry; informational.

- C.7 Execute two reads, clear Modify bit and report PTD, Set 2.
- Set RF2 to 0--0 (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set CM 1000(HEX) to A000 0000 0000 0000(HEX); clear Modify in PTD.
  - Report PTD entry; informational.

- C.8 Test Modify bit, report PTD, Set 2.
- Set CM 0 to C3--C3000000(HEX).
  - Start processor at micrand E, write word to SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 1000(HEX), report data; error if not A000 0000 0000 0000(HEX).

If Scope Mode set, go to C.0.

- C.9 CPU Status Word 1.
- Report word 1 if saved; error if reported.

- C.10 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.11 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

## SECTION 43 - PAGE MAP STATUS RAM, MODIFY BIT

This section tests the integrity of the Modify bit in all 32 locations of the Page Map Status RAM. The bit is checked to ensure it clears during a Map Purge and stays set after a Page Table Update.

Subsection 0 - Modify Bit Integrity, Map Set 0.

Subsection 1 - Modify Bit Integrity, Map Set 1.

Subsection 2 - Modify Bit Integrity, Map Set 2.

### Method:

This section completes testing of the Modify bit logic. The Modify bit in all locations and all sets is tested for data integrity.

A Page Table Descriptor (PTD) with only the Valid bit set is written to CM (1000(HEX) and reported by Condition 1. A read micrand is issued loading the corresponding Page Map location, and setting the Page Table Used bit. The PTD is again reported. A write micrand is now issued to the same address. This results in a Map hit and tests that the Map Modify bit was clear prior to the write. If the bit was set prior to the write the Modify bit in the PTD will not set. Again the PTD is reported; it now has the Valid, Used, and Modified bits set. The Modify bit is cleared in the PTD and second write micrand is issued. This tests that the Map Modify bit did set during the first write. If it did not the PTD Modify bit will reset. The PTD, showing only the Valid and Used bits, is again reported.

This sequence is repeated 32 times, once for every Map location.

### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 0.
- Set RFA to E7--E7(HEX); write data.
- Deselect all Cache sets.

### Section 43, Subsection 0

Test the Modify bit in all 32 locations of the Page Map Status RAMs in Set 0.

### Hardware Tested:

- Page Map Set 0.
- Page Map Control.

60000285 A

LMT3/P - 498

### Method:

Refer to section method.

### Test Patterns (hexadecimal):

| Pass | SVA            | Initial PTD Entry   |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | 8000 0000 0000 0000 |
| 2    | 0001 0001 0000 | 8000 1000 2000 0000 |
| 3    | 0002 0002 0000 | 8000 2000 4000 0000 |
| 4    | 0003 0003 0000 | 8000 3000 6000 0000 |
| ↓    | ↓              | ↓                   |
| V    | V              | V                   |
| 20   | 001F 001F 0000 | 8001 F003 E000 0000 |

The SVA Hash Code is always zero. Therefore, all Page Table Descriptor (PTD) entries are located at CM address 1000(HEX), which is controlled by the value of the PTA register.

The Page Frame Address in all PTDs is zero and the Page Offset part of the SVA is always zero. Therefore, the RMA for all operations is CM 0.

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| F      | Purge Cache.        |
| 12     | Purge Map.          |

### Initialization:

- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Map Set 0.
- Set Pass count to 1.

60000285 A

LMT3/P - 499

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 20(HEX)).  
a. Report pass count; informational.
- C.1 Report initial PTD.  
a. Set CM 1000(HEX) to current initial PTD.  
b. Report current initial PTD; informational.
- C.2 Execute read and report PTD.  
a. Set RF2 to current SVA.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read CM 1000(HEX), report data; error if not as reported in C.1 plus bit 2; used must be set.
- C.3 If Scope Mode set and no C.2, C.3 error, go to C.4.  
Execute write and report PTD.  
a. Set CM 0 to C3--C3000000(HEX).  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM 1000(HEX), report data; error if not as reported in C.1 plus bits 2 and 3; Modify must be set.

If Scope Mode set, go to C.0.  
If error on C.2, C.3, go to C.6.

- C.4 Clear Modify bit in PTD and report PTD.  
a. Clear bit 3 in CM; 1000(HEX).  
b. Report CM 1000(HEX), informational.
- C.5 Execute write and report PTD.  
a. Set CM 0 to C3--C3000000(HEX).  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read CM 1000(HEX), report data; error if not as reported in C.4.

If Scope Mode set, go to C.0.

- C.6 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

Conditions:

- C.7 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.8 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 20(HEX) passes complete; else advance current operand pointer and return to C.0.

Section 43, Subsection 1

Test the Modify bit in all 32 locations of the Page Map Status RAMs in Set 1.

Hardware Tested:

- Page Map Set 1.
- Page Map Control.

Method:

Refer to section method.

Test Patterns (hexadecimal):

| Pass | SVA            | Initial PTD Entry   |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | 8000 0000 0000 0000 |
| 2    | 0001 0001 0000 | 8000 1000 2000 0000 |
| 3    | 0002 0002 0000 | 8000 2000 4000 0000 |
| 4    | 0003 0003 0000 | 8000 3000 6000 0000 |
|      |                |                     |
| v    | v              | v                   |
| 20   | 001F 001F 0000 | 8001 F003 E000 0000 |

The SVA Hash Code is always zero. Therefore, all PTD entries are located at CM address 1000(HEX), which is controlled by the value of the PTA register.

The Page Frame Address in all PTDs is zero and the Page Offset part of the SVA is always zero. Therefore, the RMA for all operations is CM 0.

# Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| F      | Purge Cache.        |
| 12     | Purge Map.          |

## Initialization:

- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Map Set 1.
- Set Pass Count to 1.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 20(HEX).
  - a. Report pass count; informational.
- C.1 Report initial PTD.
  - a. Set CM 1000(HEX) to current initial PTD.
  - b. Report current initial PTD; informational.
- C.2 Execute read and report PTD.
  - a. Set RF2 to current SVA.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read CM 1000(HEX), report data; error if not as reported in C.1 plus bit 2; used must be set.

## Conditions:

- C.3 If Scope Mode set and no C.2, C.3 error, go to C.4. Execute write and report PTD.
  - a. Set CM 0 to C3--C3000000(HEX).
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 1000(HEX), report data; error if not as reported in C.1 plus bits 2; and 3 modify must be set.

If Scope Mode set, go to C.0.  
If error on C.2, C.3, go to C.6.

- C.4 Clear Modify bit in PTD and report PTD.
  - a. Clear bit 3 in CM; 1000(HEX).
  - b. Report CM 1000(HEX), informational.
- C.5 Execute write and report PTD.
  - a. Set CM 0 to C3--C3000000(HEX).
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 1000(HEX), report data; error if not as reported in C.4.

If Scope Mode set, go to C.0.

- C.6 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.7 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.8 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 20(HEX) passes complete; else advance current operand pointer and return to C.0.

## Section 43, Subsection 2

Test the Modify bit in all 32 locations of the Page Map Status RAMs in Set 2.

## Hardware Tested:

- Page Map Set 2.
- Page Map Control.



# Method:

Refer to section method.

## Test Patterns (hexadecimal):

| Pass | SVA            | Initial PTD Entry   |
|------|----------------|---------------------|
| 1    | 0000 0000 0000 | 8000 0000 0000 0000 |
| 2    | 0001 0001 0000 | 8000 1000 2000 0000 |
| 3    | 0002 0002 0000 | 8000 2000 4000 0000 |
| 4    | 0003 0003 0000 | 8000 3000 6000 0000 |
| ↓    | ↓              | ↓                   |
| 20   | 001F 001F 0000 | 8001 F003 E000 0000 |

The SVA Hash Code is always zero. Therefore, all PTD entries are located at CM address 1000(HEX), which is controlled by the value of the PTA register.

The Page Frame Address in all PTDs is zero and the Page Offset part of the SVA is always zero. Therefore, the RMA for all operations is CM 0.

## Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| F      | Purge Cache.        |
| 12     | Purge Map.          |

## Initialization:

- Purge Map All.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Select Map Set 2.
- Set Pass Count to 1.

60000285 A

LMT3/P - 504

# Conditions:

## NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 20(HEX).
  - a. Report pass count; informational.
- C.1 Report initial PTD.
  - a. Set CM 1000(HEX) to current initial PTD.
  - b. Report current initial PTD; informational.
- C.2 Execute read and report PTD.
  - a. Set RF2 to current SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read CM 1000(HEX), report data; error if not as reported in C.1 plus bit 2; used must be set.
- C.3 If Scope Mode set and no C.2, C.3 error, go to C.4.
  - Execute write and report PTD.
    - a. Set CM 0 to C3--C3000000(HEX).
    - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
    - c. Read CPU status, format and save if error; error if not 0.
    - d. Read CM 1000(HEX), report data; error if not as reported in C.1 plus bits 2 and 3; modify must be set.

If Scope Mode set, go to C.0.

If error on C.2, C.3, go to C.6.

- C.4 Clear Modify bit in PTD and report PTD.
  - a. Clear bit 3 in CM 1000(HEX).
  - b. Report CM 1000(HEX); informational.
- C.5 Execute write and report PTD.
  - a. Set CM 0 to C3--C3000000(HEX).
  - b. Start processor at micrand E, write word to SVA, wait for CPU halted.
  - c. Read CPU status, format and save if error; error if not 0.
  - d. Read CM 1000(HEX), report data; error if not as reported in C.4.

If Scope Mode set, go to C.0.

- C.6 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.

60000285 A

LMT3/P - 505

Conditions:

- C.7 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.8 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 20(HEX) passes complete; else advance current operand pointer and return to C.0.

SECTION 44 - PAGE MAP STATUS PARITY CIRCUITS (MPLRPE)

This section tests the parity generator and checker used by the Page Map LRU Code and Modify bit. The section also tests the parity error signal (MPLRPE).

Subsection 0 - Page Map Status RAM parity Logic.

Subsection 1 - Page Map Status RAM parity Signal.

Method:

Refer to subsection methods.

Initialization:

- Deselect all Cache sets.
- Select all Page Map sets.
- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Set CM; 0 through FFF(HEX), to C3--C300xxx(HEX); xxx = 0 through FFF(HEX).

Section 44, Subsection 0

Test the parity generator and checker fed by the Page Map LRU code (2 bits) and the Modify bit. Many bit combinations have been tested during previous sections. This subsection checks those bit combinations not yet used. All three sets are tested.

Hardware Tested:

- LRU/Modify Parity.

Method:

This parity logic was half tested during the Map LRU logic test; i.e., all LRU codes, with Modify bit set. This subsection will now use all LRU codes with Modify bit clear.

This is accomplished by issuing six reads, all resulting in Map misses. This should cycle counts 0 through 3 in all sets LRU update logic.

| Map Set           | LRU Code |   |   |
|-------------------|----------|---|---|
|                   | 0        | 1 | 2 |
| After Purge       | 0        | 0 | 0 |
| After first read  | 0        | 1 | 1 |
| After second read | 1        | 0 | 2 |
| After third read  | 2        | 1 | 0 |
| After fourth read | 0        | 2 | 1 |
| After fifth read  | 1        | 0 | 2 |
| After sixth read  | 2        | 1 | 0 |

To determine if the codes are as shown, all sets except 2 are disabled, and the Page Table in CM is destroyed. A seventh read is now issued using the last SVA (sixth read). A Map hit should occur indicating the LRU codes all cycled. If a Miss occurs (PWSF), Set 2 was not last allocated and an error has occurred.

#### Test Patterns (hexadecimal):

| Read | SVA            | PTD Entry           |
|------|----------------|---------------------|
| 1    | 0100 0000 0000 | E010 0000 0000 0000 |
| 2    | 0200 0000 0000 | E020 0000 0000 0000 |
| 3    | 0300 0000 0000 | E030 0000 0000 0000 |
| 4    | 0400 0000 0000 | E040 0000 0000 0000 |
| 5    | 0500 0000 0000 | E050 0000 0000 0000 |
| 6    | 0600 0000 0000 | E060 0000 0000 0000 |

#### Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| F      | Purge Cache All.    |
| 12     | Purge Map All.      |

#### Initialization:

- Purge Map All.
- Start processor at micrand 12, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).

60000285 A

LMT3/P - 508

#### Conditions:

- C.0 Report result of first read.
- a. Set RF2 to 0100 0000 0000(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Set CM 1000(HEX) to E010--0(HEX); PTD entry.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000000(HEX).
- C.1 Report result of second read.
- a. Set RF2 to 0200 0000 0000(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Set CM 1000(HEX) to E020--0(HEX); PTD entry.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000000(HEX).
- C.2 Report result of third read.
- a. Set RF2 to 0300 0000 0000(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Set CM 1000(HEX) to E030--0(HEX); PTD entry.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000000(HEX).
- C.3 Report result of fourth read.
- a. Set RF2 to 0400 0000 0000(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Set CM 1000(HEX) to E040--0(HEX); PTD entry.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000000(HEX).
- C.4 Report result of fifth read.
- a. Set RF2 to 0500 0000 0000(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Set CM 1000(HEX) to E050--0(HEX); PTD entry.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000000(HEX).
- C.5 Report result of sixth read.
- a. Set RF2 to 0600 0000 0000(HEX) (SVA).
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Set CM 1000(HEX) to E060--0(HEX); PTD entry.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000000(HEX).

60000285 A

LMT3/P - 509

# Conditions:

- C.6 Report result of final read.
  - a. Select Map Set 2 only.
  - b. Clear CM 1000(HEX), destroy PTD.
  - c. Set RF8 to DF--DF(HEX); filler.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C3000000(HEX).
- C.7 CPU Status Word 1.
  - a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.
  - a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

## Section 44, Subsection 1

This subsection tests the Page Map PE (MPLRPE) lines from LM to the PFS register in MAC.

### Hardware Tested:

- Complement Parity Bit.
- PFS82 bit 36 through 39.

### Method:

To test the MPLRPE status lines (one from each set) the corresponding Complement Parity (CPMAP) bits are set in the processor Processor Test Mode (PTM) register. This forces the parity generator to produce wrong parity.

A CPMAP bit is set in the processor PTM register and a write micrand is issued. After the operation is complete the CPU status is sampled, the corresponding MPLRPE should be set. This procedure is repeated for the remaining three sets.

### Test Patterns:

Refer to Conditions.

80000285 A

LMT3/P - 510

# Micrands Used:

| Number | Description        |
|--------|--------------------|
| E      | Write word to SVA. |

## Initialization:

- Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry.
- Set RF2 to 0--0 (SVA).
- Set RFA to E7--E7(HEX); write data.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.0, C.1, C.2 error, go to C.3. Report write data, Set 0.
    - a. Set CM 0 to C3--C3000000(HEX).
    - b. Select Map Set 0.
    - c. Set bit 23 in processor PTM register; CPMAP(0).
    - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
    - e. Read CM 0, report data; error if not E7--E7(HEX).
  - C.1 Report CPU Status Word 1, Set 0.
    - a. Read CPU status, format and report word 1; error if not 0--08000000(HEX); MPLRPE(0).
  - C.2 Report CPU Status Word 2, Set 0.
    - a. Clear bit 23.
    - b. Report word 2; error if not all 0.
- If Scope Mode set, go to C.0.  
If error on C.0, C.1, C.2, go to C.9.
- C.3 If Scope Mode set and no C.3, C.4, C.5 error, go to C.6. Report write data, Set 1.
    - a. Set CM 0 to C3--C3000000(HEX).
    - b. Select Map Set 1.
    - c. Set bit 24 in processor PTM register; CPMAP(1).
    - d. Start processor at micrand E, write word to SVA, wait for CPU halted.
    - e. Read CM 0, report data; error if not E7--E7(HEX).
  - C.4 Report CPU Status Word 1, Set 1.
    - a. Read CPU status, format and report word 1; error if not 0--04000000(HEX); MPLRPE(1).

80000285 A

LMT3/P - 511

Conditions:

- C.5 Report CPU Status Word 2, Set 1.  
a. Clear bit 24.  
b. Report word 2; error if not all 0.

If Scope Mode set, go to C.0.

If error on C.3, C.4, C.5, go to C.9.

- C.6 If Scope Mode set and no C.6, C.7, C.8 error, go to C.9.  
Report write data, Set 2.  
a. Set CM 0 to C3--C3000000(HEX).  
b. Select Map Set 2.  
c. Set bit 25 in processor PTM register; CPMAP(2).  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CM 0, report data; error if not E7--E7(HEX).
- C.7 Report CPU Status Word 1, Set 2.  
a. Read CPU status, format and report word 1; error if not 0--02000000(HEX); MPLRPE(2).
- C.8 Report CPU Status Word 2, Set 2.  
a. Clear bit 25.  
b. Report word 2; error if not all 0.

If Scope Mode set, go to C.0.

- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

SECTION 45 - LMMIC FUNCTION CODE PARITY LOGIC

This section tests the parity logic which checks the LM function code (LMMIC) sent to LM from the Segment Map. Local Memory also receives function codes from Address Control, however, all legal codes possible were sent during the ACT3/P diagnostic and they are therefore tested.

Also tested are the parity error signals, CNTLPE, which indicates a parity error in the function code or purge code was detected.

NOTE

The purge code parity checker has been tested for false parity indications. This testing was implicitly performed during the many subsections testing the various purge functions. All possible purge codes have been used.

Subsection 0 - LMMIC Parity Checker.

Subsection 1 - CNTLP Parity Error Signals.

Method:

Refer to subsection methods.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).

Section 45, Subsection 0

Test the Local Memory Micro Code (LMMIC) parity checkers. Successful completion indicates no false parity errors are generated.

Hardware Tested:

- LMMIC Parity Checkers.

Method:

The LMMIC parity checkers test bits 18 and 20 through 29 of the Functional Unit (FU) Micrand. Two circuits are used to test these bits.

- Chip one tests bits 18 and 20 through 23; 5 bits.
- Chip two tests bits 24 through 29; 6 bits.

By using a double incrementing pattern counting from 0 through 3F(HEX) (8 bits) both chips are tested simultaneously. The double pattern is formed and positioned in the LMMIC field of the FU Micrand and a write type micrand is issued.

LMMIC

00x0 xxxx yyyy yy00  
16 31

x and y represent bits checked by chips one and two. This also shows the placement of the double pattern see Test Patterns for additional clarity). Note bit 19 must be cleared after pattern formation.

Most LMMIC values used in this subsection are not normally used, and some are even illegal. However, the purpose is to sense false parity indications; other results are ignored.

Micrand E is read from Control Store and used as the base micrand. LMMIC is cleared and the current pattern inserted and used, updated and used again, etc.

Master Clear is issued between passes to eliminate any problem encountered with illegal functions.

Test Patterns (hexadecimal):

| Pass | Pattern             |
|------|---------------------|
| 1    | 0000                |
| 2    | 0104                |
| 3    | 0208                |
| 4    | 030C                |
| 5    | 0410                |
|      |                     |
| v    | v                   |
| 10   | 0F3C                |
| 11   | 0040 bit 19 cleared |
| 12   | 0144                |
|      |                     |
| v    | v                   |
| 1F   | 0E78                |
| 20   | 0F7C                |
| 21   | 2080                |
|      |                     |
| v    | v                   |
| 3F   | 2EF8                |
| 40   | 2FFC                |

Micrands Used:

A modified version of Micrand E is used.

Initialization:

- Read Micrand E from Control Store and clear the LMMIC field.
- Set RF2 to 0--0(HEX) (SVA).
- Set Pass Count to 1.

Conditions:

- C.0 Pass count (1 through 40(HEX).
  - a. Report pass count; informational.
- C.1 Report current FU micrand.
  - a. Write base micrand to Control Store work area.
  - b. Report FU portion of base micrand; informational.
- C.2 Report CPU Status Word 1.
  - a. Start processor at Control Store Work Area, wait for CPU halted or time out.
  - b. Read CPU status, format and report word 1; error if CNTLPE bits not 0.
- C.3 Report CPU Status Word 2.
  - a. Report word 2; informational.
- C.4 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 40(HEX) passes complete; else advance LMMIC pattern and return to C.0.

Section 45, Subsection 1

This subsection tests the parity error signals and status bits which indicate a function code parity error has occurred. The function code can be the Local Memory Micrand (LMMIC) field of the FU micrand, the Address Control function code (ACFN) that is generated and sent from Address Control, or the Purge function code. The purge parity indication is not tested by LMT3/P.

Hardware Tested:

- LMMIC PE Signals.
- ACFN PE Signal.
- ACFN PE Force Bit.
- PFS82 Bits 33, 34, and 35.

Method:

There are three CNTLPE bits. Bit 0 indicates a Purge code parity error and is tested by the Segment Map Test SMT3/P.

Bits 1 and 2 indicate a parity error in the LMMIC function code. These bits are tested by forcing wrong parity in the LMMIC field in Control Store. When the micrand is issued a parity error should be detected and the status bit set.

The ACFN Code does not have a parity bit.

Test Patterns:

Scratch Micrand = 0060 66B2 0041 8250 0900 0000 4800 0000.

(All bytes contain even parity).

Micrands Used:

| Number | Description                            |
|--------|----------------------------------------|
| 3B     | Even parity micrand for LMMIC testing. |

Initialization:

- Set RF2 to 0--0(HEX) (SYA).
- Set CM 1000(HEX) to B0--0(HEX); PTD entry.

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.0, C.1 error, go to C.2.  
Report CPU Status Word 1 after CNTLPE (1) test.
- a. Set LMMIC field in scratch micrand to 0100(HEX).
  - b. Set bit 18 in processor PTM register, MAC data output bit.
  - c. Write scratch micrand to Control Store work area.
  - d. Clear bit 18 in processor PTM register.
  - e. Start processor at Control Store work area, wait for CPU halted.
  - f. Read CPU status, format and report word 1; error if not 0--040000000(HEX); CNTLPE (1).

Conditions:

- C.1 Report CPU Status Word 2.  
a. Report word 2; error if not 0.

If Scope Mode set, go to C.0.

- C.2 Report CPU Status Word 1 after CNTLPE (2) Test.
- a. Set LMMIC field in scratch micrand to 0010(HEX).
  - b. Set bit 18 in processor PTM register, MAC data output bit.
  - c. Write scratch micrand to Control Store work area.
  - d. Clear bit 18 in processor PTM register.
  - e. Start processor at Control Store work area, wait for CPU halted.
  - f. Read CPU status, format and report word 1; error if not 0--020000000(HEX); CNTLPE (2).

- C.3 Report CPU Status Word 2.  
a. Report word 2; error if not 0.

- C.4 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

If Scope Mode set, go to C.0.

Clear Control Store Work Area.

SECTION 46 - TEST THE CAR EQUAL (FESE) COMPARATOR

This section tests both uses of the CACHE ADDRESS REGISTER (CAR) and the CAR Equal Compare (FESE) logic. It is used to:

- 1. Delay a read request when it follows a write request to the same word in Cache. The read must be delayed until the write cycle has updated Cache.
- 2. Bypass the Cache LRU Code around the Tag RAM when two requests are issued contiguously to the same Cache Block.

Subsection 0 - FESE Comparator, Write/Read Delay, Compare Valid.

Subsection 1 - FESE Comparator, Write/Read Delay, Compare Invalid.

Subsection 2 - FESE Comparator, LRU Shortstop.

Method:

Refer to subsection methods.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Select all Map Sets.
- Select all available Cache Sets.

Section 46, Subsection 0

This subsection tests the 10-bit CAR Equal Compare (FESE) Comparator when initiating a write followed by read sequence to the same Cache word. This subsection tests for valid compares only.

Hardware Tested:

- FESE Comparator.
- Write Conflict Logic.

Method:

When issuing a write micrand followed immediately by a read micrand to the same word and that word is in Cache, the read must be delayed until the new data is stored in Cache. If the comparison fails, the read is allowed to read the old or original data.

This subsection uses a double micrand sequence of write (II Request) and read (II Request) to the same SVA. SVAs of zeros, ones and alternate bits are used to ensure all bits in the 10-bit comparator detect compares.

The same test situation is issued twice; once in normal mode and once in FAKECM mode because the delay must be invoked in either case.

Test Patterns (hexadecimal):

| Conditions | SVA            | PTD Entry           |
|------------|----------------|---------------------|
| 0, 1, 2    | 0000 0000 0000 | B000 0000 0000 0000 |
| 3, 4, 5    | 0000 0000 1FF8 | B000 0000 03C0 0000 |
| 6, 7, 8    | 0000 0000 1550 | B000 0000 0280 0000 |
| 9, 10, 11  | 0000 0000 0AA8 | B000 0000 0140 0000 |

Micrands Used:

| Number | Description                                     |
|--------|-------------------------------------------------|
| D      | Read Word from SVA.                             |
| 29     | Write/Read Sequence using SVAs; II/II Requests. |
| F      | Purge Cache All.                                |

Initialization:

- Set PSM to 0.
- Set CM 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set RF13 to 0; Micrand 29 loop count.

Conditions:

NOTE

Disregard skipped conditions on error display.



Conditions:

- C.0 If Scope Mode set and no C.0, C.1, C.2 error, go to C.3.  
Block fill block 0, report data read.  
a. Set RF2 to 0--0(HEX) (SVA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD entry.  
d. Set CM 0 to C3--C3000000(HEX).  
e. Start processor at micrand D, read word from SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF6, report data; error if not C3--C3000000(HEX).

If Scope Mode set and no C.0, C.1 error, go to C.2.

- C.1 Execute Normal mode test, report read data.  
a. Set RFA to E7--E7(HEX); write data.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set RF3 to 0--0(HEX); Read SVA.  
d. Start processor at micrand 29, write/read sequence using SVAs; 11/11 Requests, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF6, report data; error if not E7--E7(HEX).

If Scope Mode set, go to C.0.

If error on C.0, C.1, go to C.12.

- C.2 Execute FAKECM mode test, report read data.  
a. Set RFA to C3--C3000000(HEX); write data.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Select FAKECM mode.  
d. Start processor at micrand 29, write/read sequence using SVAs; 11/11 Requests, wait for CPU halted.  
e. Deselect FAKECM mode.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF6, report data; error if not C3--C3000000(HEX).

If Scope Mode set, go to C.0.

If error on C.2, go to C.12.

- C.3 If Scope Mode set and no C.3, C.4, C.5 error, go to C.6.  
Block fill block FF(HEX), report data read.  
a. Set RF2 to 0--01FF8(HEX) (SVA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set CM 1000(HEX) to B000 0000 03C0 0000(HEX); PTD entry.  
d. Set CM; (3FF(HEX)), to C3--C30003FF(HEX).  
e. Start processor at micrand D, read word from SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF6, report data; error if not C3--C30003FF(HEX).

If Scope Mode set and no C.3, C.4 error, go to C.5.

Conditions:

- C.4 Execute Normal mode test, report read data.  
a. Set RFA to E7--E7(HEX); write data.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set RF3 to 0--01FF8(HEX); Read SVA.  
d. Start processor at micrand 29, write/read sequence using SVAs; 11/11 Requests, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF6, report data; error if not E7--E7(HEX).

If Scope Mode set, go to C.0.

If error on C.3, C.4, go to C.12.

- C.5 Execute FAKECM mode test, report read data.  
a. Set RFA to C3--C30003FF(HEX); write data.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Select FAKECM mode.  
d. Start processor at micrand 29, write/read sequence using SVAs; 11/11 Requests, wait for CPU halted.  
e. Deselect FAKECM mode.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF6, report data; error if not C3--C303FF(HEX).

If Scope Mode set, go to C.0.

If error on C.5, go to C.12.

- C.6 If Scope Mode set and no C.6, C.7, C.8 error, go to C.9.  
Block fill block AA(HEX), report data read.  
a. Set RF2 to 0--01550(HEX) (SVA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set CM 1000(HEX) to B000 0000 0280 0000(HEX); PTD entry.  
d. Set CM; (2AA(HEX)), to C3--C30002AA(HEX).  
e. Start processor at micrand D, read word from SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF6, report data; error if not C3--C30002AA(HEX).

- C.7 If Scope Mode set and no error on C.6, C.7, go to C.8.  
Execute Normal mode test, report read data.  
a. Set RFA to E7--E7(HEX); write data.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Set RF3 to 0--01550(HEX); read SVA.  
d. Start processor at micrand 29, write/read sequence using SVAs; 11/11 Requests, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF6, report data; error if not E7--E7(HEX).

If Scope Mode set, go to C.0.

If error on C.6, C.7, go to C.12.

Conditions:

- C.8      Execute FAKECM mode test, report read data.  
a. Set RFA to C3--C30002AA(HEX); write data.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Select FAKECM mode.  
d. Start processor at micrand 29, write/read sequence using SVAs; II/II Requests, wait for CPU halted.  
e. Deselect FAKECM mode.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF8, report data; error if not C3--C30002AA(HEX).

If Scope Mode set, go to C.0.  
If error on C.8, go to C.12.

- C.9      Block fill block 55(HEX), report data read.  
a. Set RF2 to 0--0AA8(HEX) (SVA).  
b. Set RF8 to DF--DF; filler.  
c. Set CM 1000(HEX) to B000 0000 0140 0000(HEX); PTD entry.  
d. Set CM; (155(HEX)), to C3--C3000155(HEX).  
e. Start processor at micrand D, read word from SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF8, report data; error if not C3--C3000155(HEX).

- C.10     If Scope Mode set and no C.9, C.10 error, go to C.11.  
Execute Normal mode test, report read data.  
a. Set RFA to E7--E7(HEX); write data.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Set RF3 to 0--0AA8(HEX); read SVA.  
d. Start processor at micrand 29, write/read sequence using SVAs; II/II Requests, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF8, report data; error if not E7--E7(HEX).

If Scope Mode set, go to C.0.  
If error on C.9, C.10, go to C.12.

- C.11     Execute FAKECM mode test, report read data.  
a. Set RFA to C3--C3000155(HEX); write data.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Select FAKECM mode.  
d. Start processor at micrand 29, write/read sequence using SVAs; II/II Requests, wait for CPU halted.  
e. Deselect FAKECM mode.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF8, report data; error if not C3--C3000155(HEX).

If Scope Mode set, go to C.0.

- C.12     CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

- C.13     CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

Conditions:

- C.14     First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Section 46, Subsection 1

This subsection tests the 10-bit CAR Equal Compare (FESE) Comparator for false compare indications.

Hardware Tested:

- FESE Comparator.

Method:

The same 2-micrand sequence used in subsection 0 is used again here. In this subsection, however, the write and read addresses are different by one bit. If the comparator fails in that bit position a false compare occurs. If a false compare occurs the only indication would be the length of time required to complete the sequence. The Write Delay logic would be invoked causing a major cycle delay. The write and read addresses actually sent to Cache would be good. Therefore, the data written and read would reflect no error.

Because timing is the only indication, each test sequence is repeated 32 times before allowing the CPU to halt. This allows the timing discrepancy to build to a level of detection. If the total time taken to execute the 32 sequences is not within one of the expected results, an error is reported. The CMC Free Running Counter is used to time the test sequence.

The patterns used consist of a sliding one bit and a sliding zero bit passed through all ten comparator bit positions.

Test Patterns (hexadecimal):

| Pass | Positive<br>Read SVA | Positive<br>PTD Entry | Positive<br>RMA |
|------|----------------------|-----------------------|-----------------|
| 1    | 0000 0000 0008       | B000 0000 0000 0000   | 000001          |
| 2    | 0000 0000 0010       | B000 0000 0000 0000   | 000002          |
| 3    | 0000 0000 0020       | B000 0000 0000 0000   | 000004          |
| 4    | 0000 0000 0040       | B000 0000 0000 0000   | 000008          |
| 5    | 0000 0000 0080       | B000 0000 0000 0000   | 000010          |
| 6    | 0000 0000 0100       | B000 0000 0000 0000   | 000020          |
| 7    | 0000 0000 0200       | B000 0000 0000 0000   | 000040          |
| 8    | 0000 0000 0400       | B000 0000 0000 0000   | 000080          |
| 9    | 0000 0000 0800       | B000 0000 0000 0000   | 000100          |
| A    | 0000 0000 1000       | B000 0000 0000 0000   | 000200          |

Positive Write SVA is always 0--0.

| Pass | Complement<br>Read SVA | Complement<br>PTD Entry | Complement<br>RMA |
|------|------------------------|-------------------------|-------------------|
| 1    | FFFF 7FFF FFF0         | BFFF FFFF F000 0000     | 000FFE            |
| 2    | FFFF 7FFF FFE8         | BFFF FFFF F000 0000     | 000FFD            |
| 3    | FFFF 7FFF FFD8         | BFFF FFFF F000 0000     | 000FFB            |
| 4    | FFFF 7FFF FF88         | BFFF FFFF F000 0000     | 000FF7            |
| 5    | FFFF 7FFF FF78         | BFFF FFFF F000 0000     | 000FEF            |
| 6    | FFFF 7FFF FEF8         | BFFF FFFF F000 0000     | 000FDF            |
| 7    | FFFF 7FFF FDF8         | BFFF FFFF F000 0000     | 000FBF            |
| 8    | FFFF 7FFF FBF8         | BFFF FFFF F000 0000     | 000F7F            |
| 9    | FFFF 7FFF F7F8         | BFFF FFFF F000 0000     | 000EFF            |
| A    | FFFF 7FFF EFF8         | BFFF FFFF F000 0000     | 000DFF            |

Complement Write SVA is always FFFF 7FFF FFF8.

#### Micrands Used:

| Number | Description                            |
|--------|----------------------------------------|
| D      | Read Word from SVA.                    |
| 29     | Write/Read using SVAs; II/II Requests. |
| F      | Purge Cache All.                       |

#### Initialization:

- Set RF13 to 1F(HEX); Micrand 29 loop count -1.
- Set PSM to 40(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set Pass Count to 1.
- Set RFA to E7--E7(HEX); write data.

#### Conditions:

##### NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.1, C.2, C.3, C.4 error, go to C.5.  
Pass count (1 through A(HEX)).  
a. Report pass count; informational.

80000285 A

LMT3/P - 524

#### Conditions:

- C.1 Block fill read block, report read data.  
a. Set RF2 to current positive read SVA.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Set CM 1000(HEX) to current positive PTD entry.  
d. Start processor at micrand D, read word from SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF8, report data; error if not C3--C30xxx(HEX); where xxx = current positive RMA.
- C.2 Report positive read SVA.  
a. Report current positive read SVA; informational.
- C.3 Report positive write SVA.  
a. Set RF3 to 0--0; positive write SVA.  
b. Report 0s; informational.
- C.4 Execute test, report CPU major cycles.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Clear the CMC Free Running Counter.  
c. Start processor at micrand 29, write/read sequence using SVAs; II/II Requests, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report CPU time in major cycles; error if not 1C(HEX) or 1D(HEX).

If Scope Mode set, go to C.0.

If error on C.1, C.2, C.3, C.4, go to C.5.

- C.5 Block fill read block, report read Data.  
a. Set RF2 to current complement read SVA.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Set CM 1000(HEX) to current complement PTD entry.  
d. Start processor at micrand D, read word from SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Read RF8, report data; error if not C3--C30xxx(HEX); where xxx = current complement RMA.
- C.6 Report complement read SVA.  
a. Report current complement read SVA; informational.
- C.7 Report complement write SVA.  
a. Set RF3 to FFFF 7FFF FFF8(HEX); complement write SVA.  
b. Report 0s; informational.
- C.8 Execute test, report CPU major cycles.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Clear the CMC Free Running Counter.  
c. Start processor at micrand 29, write/read sequence using SVAs; II/II Requests, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Report CPU time in major cycles; error if not 1A(HEX) or 1F(HEX).

80000285 B

LMT3/P - 525

# Conditions:

If Scope Mode set, go to C.0.

C.9 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

C.10 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

C.11 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-83; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if A(HEX) passes complete; else advance current operand pointer and return to C.0.

## Section 46, Subsection 2

Test the Cache LRU Code path around the Cache Status RAM. This path is used when two Cache hits occur in the same block one major cycle apart.

## Hardware Tested:

- Cache Tag Status.
- FESE Block Compare Output.

## Method:

The LRU Shortstop path is tested using the same method used originally to test the LRU Code Update Network. If the shortstop path should fail the problem would appear as a bad or unexpected LRU Code.

Four read operations are issued using different SVAs but all to the same Cache block. All result in an allocate and block fill operation. This sets up the LRU Codes for the actual shortstop test. The test is performed by executing two additional reads one major cycle apart. Both are again to the same block, but both result in Cache hits. The FESE block compare is sensed and the LRU Code is gated around the RAM between the two reads.

To determine if the LRU Code was correctly updated, a seventh read to the same block is issued. This read results in a Cache miss causing an allocate and block fill. All Cache sets are disabled except the set expected to have been allocated for the seventh read. FAKECM mode is used to read word 0 from Cache. If the data is from the last block fill, the correct set was allocated and the LRU Codes were properly updated.

This process is repeated, starting with reads five and six (the double read), three additional times if four sets are available. If only two sets are available parts of the test are skipped. Refer to the following chart for details.

## Operational Sequence and Expected LRU Codes:

| Event                  | 4 Sets |   |   |   | 2 Sets |   |
|------------------------|--------|---|---|---|--------|---|
|                        | 0      | 1 | 2 | 3 | 0      | 1 |
| LRU After Purge        | 0      | 0 | 0 | 0 | 0      | 0 |
| LRU After first read   | 0      | 1 | 1 | 1 | 0      | 1 |
| LRU After second read  | 1      | 0 | 2 | 2 | 1      | 0 |
| LRU After third read   | 2      | 1 | 0 | 3 |        |   |
| LRU After fourth read  | 3      | 2 | 1 | 0 |        |   |
| LRU After fifth read   | 0      | 3 | 2 | 1 | 0      | 1 |
| LRU After sixth read   | 1      | 0 | 3 | 2 | 1      | 0 |
| LRU After seventh read | 2      | 1 | 0 | 3 | 0      | 1 |
| LRU After fifth read   | 3      | 2 | 1 | 0 | 1      | 0 |
| LRU After sixth read   | 0      | 3 | 2 | 1 | 0      | 1 |
| LRU After seventh read | 1      | 0 | 3 | 2 | 1      | 0 |
| LRU After fifth read   | 2      | 1 | 0 | 3 |        |   |
| LRU After sixth read   | 3      | 2 | 1 | 0 |        |   |
| LRU After seventh read | 0      | 3 | 2 | 1 |        |   |
| LRU After fifth read   | 1      | 0 | 3 | 2 |        |   |
| LRU After sixth read   | 2      | 1 | 0 | 3 |        |   |
| LRU After seventh read | 3      | 2 | 1 | 0 |        |   |

The BKUSED bit, bit 2 of the Status RAM, is also shortstopped, but it is not tested. A failure to shortstop the BKUSED bit would have no affect.

## Test Patterns (hexadecimal):

|              | SVa            | PTD Entry           |
|--------------|----------------|---------------------|
| First read   | 0000 0000 0000 | F000 0000 0000 0000 |
| Second read  | 0100 0000 0000 | F010 0000 0000 0001 |
| Third read   | 0200 0000 0000 | F020 0000 0000 0002 |
| Fourth read  | 0300 0000 0000 | F030 0000 0000 0003 |
| Seventh read | 0700 0000 0000 | B070 0000 0000 0007 |

The fifth and sixth reads use two of the first four SVAs.

CM is initialized with the word address pattern; for example, C3C3 C3C3 C300 0xxx(HEX), xxx is set to the word address.

# Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| D      | Read Word from SVA.                        |
| F      | Purge Cache All.                           |
| 1A     | Double read from two SVAs; II/II Requests. |

## Initialization:

- Set PSM to 7F(HEX).
- Purge Cache All.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM; 0 through FFF(HEX) to C3-C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Set up Page Table starting at CM; 1000(HEX), five entries.
- Select all Cache Sets.

## Conditions:

### NOTE

Disregard skipped conditions on error display.

- C.0 Initialize block 0 in all Cache Sets.
- a. Set RF2 to 0--0(HEX); first read SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8; error if not C3--C3000000(HEX).
  - f. If error report data and exit condition.
  - g. Set RF2 to 0100 0000 0000(HEX); second read SVA.
  - h. Set RF6 to DF--DF(HEX); filler.
  - i. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - j. Read CPU status, format and save if error; error if not 0.
  - k. Read RF8; error if not C3--C3000040(HEX).
  - l. If error report data and exit condition.
  - m. If only 2 sets available, report 0--02 and exit condition; informational.
  - n. Set RF2 to 0200 0000 0000(HEX); third read SVA.
  - o. Set RF6 to DF--DF(HEX); filler.
  - p. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - q. Read CPU status, format and save if error; error if not 0.
  - r. Read RF8; error if not C3--C3000080(HEX).
  - s. If error report data and exit condition.
  - t. Set RF2 to 0300 0000 0000(HEX); fourth read SVA.
  - u. Set RF6 to DF--DF(HEX); filler.

80000285 A

LMT3/P - 528

## Conditions (C.0 continued):

- v. Start processor at micrand D, read word from SVA, wait for CPU halted.
- w. Read CPU status, format and save if error; error if not 0.
- x. Read RF8; error if not C3--C30000C0(HEX).
- y. If error, report data and exit condition.
- z. Report 0--04; informational.

Test 1, report data from sixth read.

- a. Set RF2 to 0000 0000 0000(HEX); fifth read SVA.
- b. Set RF3 to 0100 0000 0000(HEX); sixth read SVA.
- c. Set RF6 to DF--DF(HEX); filler
- d. Set RF7 to DF--DF(HEX); filler
- e. Start processor at micrand 1A, double read from two SVAs; II/II requests, wait for CPU halted.
- f. Read CPU status, format and save if error; error if not 0.
- g. Read RF8; error if not C3--C3000000(HEX). If error report data and exit condition.
- h. Read RF7, report data; error if not C3--C3000040(HEX).

## C.2

Test 1, report data from seventh read.

- a. Set RF2 to 0700 0000 0000(HEX); seventh read SVA.
- b. Set RF6 to DF--DF(HEX); filler.
- c. Start processor at micrand D, read word from SVA, wait for CPU halted.
- d. Read CPU status, format and save if error; error if not 0.
- e. Read RF8, report data; error if not C3--C30001C0(HEX).

## C.3

Test 1, report data from expected Set.

- a. Select Cache Set 2 if 4 cache sets available; else select set 0.
- b. Select FAKECM mode.
- c. Set RF6 to DF--DF(HEX); filler.
- d. Start processor at micrand D, read word from SVA, wait for CPU halted.
- e. Read CPU status, format and save if error; error if not 0.
- f. Deselect FAKECM mode.
- g. Read RF8, report data; error if not C3--C30001C0(HEX).

If Scope Mode set, go to C.0.

If error on C.0, C.1, C.2, C.3, go to C.13.

## C.4

If Scope Mode set and no C.4, C.5, C.6 error, go to C.7.

Test 2, report data from sixth read.

- a. Set RF2 to 0200 0000 0000(HEX) if four sets available; else set RF2 to 0--0 SVA.
- b. Start processor at micrand D, read word from SVA, wait for CPU halted.
- c. Read CPU status, format and save if error; error if not 0.
- d. Select all Cache sets.
- e. If two sets, set RF2 to 0100 0000 0000(HEX); fifth read SVA.
- f. If four sets, set RF2 to 0300 0000 0000(HEX); fifth read SVA.
- g. Set RF3 to 0000 0000 0000(HEX); sixth read SVA.
- h. Set RF6 to DF--DF(HEX); filler.

80000285 A

LMT3/P - 529

Conditions (C.4 continued):

- i. Set RF7 to DF--DF(HEX); filler.
- j. Start processor at micrand 1A, double read from two SVAs; 11/11 requests, wait for CPU halted.
- k. Read CPU status, format and save if error; error if not 0.
- l. Read RF8; error if not:  
C3--C3000040(HEX); if 2 sets,  
C3--C30000C0(HEX); if 4 sets,
- m. If error, report data and exit condition.
- n. Read RF7, report data; error if not C3--C3000000(HEX).

- C.5 Test 2, report data from seventh read.
- a. Set RF2 to 0700 0000 0000(HEX); seventh read SVA.
  - b. Set RF6 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not C3--C30001C0(HEX).

- C.6 Test 2, report data from expected Set.
- a. Select Cache Set 1 only.
  - b. Select FAKECM mode.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Deselect FAKECM mode.
  - g. Read RF8, report data; error if not C3--C30001C0(HEX).

If Scope Mode set, go to C.0.  
If error on C.4, C.5, C.6, go to C.13.

- C.7 If Scope Mode set and no C.7, C.8, C.9 error, go to C.10.
- Test 3, report data from sixth read.
- a. If only two sets available, report 0s and exit condition.
  - b. Set RF2 to 0100 0000 0000(HEX); reinitialize set 1.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Select all Cache sets.
  - f. Set RF2 to 0200 0000 0000(HEX); fifth read SVA.
  - g. Set RF3 to 0300 0000 0000(HEX); sixth read SVA.
  - h. Set RF6 to DF--DF(HEX); filler.
  - i. Set RF7 to DF--DF(HEX); filler.
  - j. Start processor at micrand 1A, double read from two SVAs; 11/11 requests, wait for CPU halted.
  - k. Read CPU status, format and save if error; error if not 0.
  - l. Read RF8; error if not C3--C3000080(HEX). If error report data and exit condition.
  - m. Read RF7, report data; error if not C3--C30000C0(HEX).

Conditions:

- C.8 Test 3, report data from seventh read.
- a. If only two sets available, report 0s and exit condition.
  - b. Set RF2 to 0700 0000 0000(HEX); seventh read SVA.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C30001C0(HEX).
- C.9 Test 3, report data from expected Set.
- a. If only 2 sets available, report 0s and exit condition.
  - b. Select Cache Set 0 only.
  - c. Select FAKECM mode.
  - d. Set RF6 to DF--DF(HEX); filler.
  - e. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - f. Read CPU status, format and save if error; error if not 0.
  - g. Deselect FAKECM mode.
  - h. Read RF8, report data; error if not C3--C30001C0(HEX).

If Scope Mode set, go to C.0.  
If error on C.7, C.8, C.9, go to C.13.

- C.10 Test 4, report data from sixth read.
- a. If only 2 sets available, report 0s and exit condition.
  - b. Set RF2 to 0000 0000 0000(HEX); reinitialize set 0.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Select all Cache sets.
  - f. Set RF2 to 0100 0000 0000(HEX); fifth read SVA.
  - g. Set RF3 to 0200 0000 0000(HEX); sixth read SVA.
  - h. Set RF6 to DF--DF(HEX); filler.
  - i. Set RF7 to DF--DF(HEX); filler.
  - j. Start processor at micrand 1A, double read from two SVAs; 11/11 requests, wait for CPU halted.
  - k. Read CPU status, format and save if error; error if not 0.
  - l. Read RF8; error if not C3--C3000040(HEX). If error report data and exit condition.
  - m. Read RF7, report data; error if not C3--C3000080(HEX).
- C.11 Test 4, report data from seventh read.
- a. If only two sets available, report 0s and exit condition.
  - b. Set RF2 to 0700 0000 0000(HEX); seventh read SVA.
  - c. Set RF6 to DF--DF(HEX); filler.
  - d. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - e. Read CPU status, format and save if error; error if not 0.
  - f. Read RF8, report data; error if not C3--C30001C0(HEX).

Conditions:

- C.12 Test 4, report data from expected Set.
- a. If only two sets available, report 0s and exit condition.
  - b. Select Cache Set 3 only.
  - c. Select FAKECM mode.
  - d. Set RF8 to DF--DF(HEX); filler.
  - e. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - f. Read CPU status, format and save if error; error if not 0.
  - g. Deselect FAKECM mode.
  - h. Read RF8, report data; error if not C3--C30001C0(HEX).

If Scope Mode set, go to C.0.

- C.13 CPU Status Word 1.
- a. Report word 1 if saved; error if reported.
- C.14 CPU Status Word 2.
- a. Report word 2 if saved; error if reported.
- C.15 First Failure Capture Information.
- a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

SECTION 47 - CONFLICT TESTING, ALLOCATION DELAYS

This section tests conflict situations which result in delaying Cache allocation.

A Wait Allocate occurs when a Block Fill requests Cache allocation during the same minor cycle that a data word is received from CM due to a previous Block Fill. The word coming from CM must be stored in Cache, the allocation must wait for an idle minor cycle.

Method:

Refer to subsection methods.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Select all Map Sets.
- Select all Available Cache Sets.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).

Section 47, Subsection 0

This subsection tests the Wait Allocate logic. Wait Allocate occurs when a Block Fill requests Cache allocation during the same major cycle that a data word is received from CM due to a previous Block Fill. The word coming from CM is stored in Cache, the word valid allocation waits for the next idle major cycle.

Hardware Tested:

- Wait Allocate Network.

Method:

To create the situation described above, two contiguous reads, both from Instruction Issue (II), are issued. Both reads result in a Cache miss and, therefore, request a Block Fill. The first Block Fill request is sent to CM without delay. However, word valid allocation from the second request must wait for the data from the first request to be received.

When the first word of the block (first request) is received by LM, the word and response are sent to II. This allows the second read to be issued. The second request is accepted and a Block Fill is initiated. This Block Fill allocation should be delayed because the remaining words from the first block are still being received and stored in Cache.

Two variations are used to test the Wait Allocate conflict. Pass one uses two read requests issued from II. Pass two issues the first request from II and the second request from Instruction Fetch (IF). To create the wait allocate conflict using an IF request, the IF (set P) micrand must be issued first. The II micrand is then issued at such a time that the second IF request will have to wait for the II Block Fill.

A failure can manifest itself in two ways:

1. The CPU can hang, causing the micrand to time out.
2. Data received and stored in Cache will not be as expected.

Both situations are sensed for and reported.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| F      | Purge Cache All.                           |
| D      | Read Word from SVA.                        |
| 12     | Purge Page Maps.                           |
| 1A     | Double Read from two SVAs; II/II requests. |
| 3D     | Double Read from two SVAs; IF/II requests. |

Initialization:

- Purge Page Maps.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU Status, format and save is error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD for first read.
- Set P Descriptor register to zero.
  - Set RF11 to zero; segment number.
  - Set CM 0 to AA0--0(HEX); P Descriptor value.
  - Start processor at micrand 10, Wait for CPU halted.
  - Read CPU Status, format and save is error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set pass count to 1.
- Set RF2 to 0--0; SVA for first read.
- Set RF3 to 0--040(HEX); SVA for second read, pass 1.
- Set RF10 to 0-038(HEX); PVA for second read, pass 2.

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 or 2).
- a. Set RF2 to 0--0(HEX) (SVA).
  - b. Report pass count; informational.
- C.1 Execute test, report CPU hang status.
- a. Start processor at micrand F, purge Cache all, wait for CPU halted.
  - b. Read CPU Status, format and save if error; error if not 0.
  - c. Set CM 0 to C3--C3000000(HEX).
  - d. Set RF8 to DF--DF(HEX); filler.
  - e. Set RF7 to DF--DF(HEX); filler.
  - f. Set RF14 to DF--DF(HEX); filler.
  - g. If Pass 1, start processor at micrand 1A, II/II read words from 2 SVAs, wait for CPU halted.
  - h. If Pass 2, start processor at micrand 3D, IF/II requests, wait for CPU halted.
  - i. Read CPU status, format and save if error; error if not 0. If bit 16; CPU hang is set report 0--01(HEX); else report 0.
- C.2 If Scope Mode set and no C.1, C.2 error, go to C.3. Report word 0 from first read.
- a. Select FAKECM mode.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not C3--C3000000(HEX).

If Scope Mode set, go to C.0.  
If error on C.1, C.2, go to C.10.

- C.3 If Scope Mode set and no C.3 error, go to C.4. Report word 1 from first read.
- a. Set RF2 to 0000 0000 0008(HEX); SVA, word 1.
  - b. Set RF8 to DF--DF(HEX); filler.
  - c. Start processor at micrand D, read word from SVA, wait for CPU halted.
  - d. Read CPU status, format and save if error; error if not 0.
  - e. Read RF8, report data; error if not C3--C3000001(HEX).

If Scope Mode set, go to C.0.  
If error on C.3, go to C.10.



Conditions:

- C.4 If Scope Mode set and no C.4 error, go to C.5.  
Report word 2 from first read.  
a. Set RF2 to 0000 0000 0010(HEX); SVA, word 1.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000002(HEX).

If Scope Mode set, go to C.0.

If error on C.4, go to C.10.

- C.5 If Scope Mode set and no C.5 error, go to C.6.  
Report word 3 from first read.  
a. Set RF2 to 0000 0000 0018(HEX); SVA, word 1.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000003(HEX).

If Scope Mode set, go to C.0.

If error on C.5, go to C.10.

- C.6 If Scope Mode set and no C.6 error, go to C.7.  
Report word 0 from second read.  
a. Set RF2 to 0--040(HEX); SVA, word 0.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000008(HEX).

If Scope Mode set, go to C.0.

If error on C.6, go to C.10.

- C.7 If Scope Mode set and no C.7 error, go to C.8.  
Report word 1 from second read.  
a. Set RF2 to 0--048(HEX); SVA, word 1.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000009(HEX).

If Scope Mode set, go to C.0.

If error on C.7, go to C.10.

Conditions:

- C.8 If Scope Mode set and no C.8 error, go to C.9.  
Report word 2 from second read.  
a. Set RF2 to 0--050(HEX); SVA, word 1.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C300000A(HEX).

If Scope Mode set, go to C.0.

If error on C.8, go to C.10.

- C.9 Report word 3 from second read.  
a. Set RF2 to 0--058(HEX); SVA, word 3.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Deselect FAKECM mode.  
f. Read RF6, report data; error if not C3--C300000B(HEX).

If Scope Mode set, go to C.0.

- C.10 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

- C.11 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

- C.12 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if two passes complete; else return to C.0.

## SECTION 48 - CACHE/PAGE MAP CONFLICTS

Several situations can occur when two LM micrands are issued in sequence, depending on the state of the Page Map and Cache. If both hit, one thing happens. If only one hits, different circuitry is used, and so forth. The end result, however, should be the same. This section covers the various Cache/Page Map Conflicts.

Subsection 0 - Cache to Map Conflict (CHTMP).

Subsection 1 - Read Access and Read/Write Conflict.

Subsection 2 - Cache Allocation after Word Not Valid Hit.

Subsection 3 - Page Table Update Conflicts.

Method:

Refer to subsection methods.

Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 7F(HEX).
- Select all Map Sets.
- Select all Available Cache Sets.

Section 48, Subsection 0

Test the conflict situation where two read operations are issued, one following the other, from different sources. Both reads result in a Cache miss. This situation causes a MAPOUT condition.

Hardware Tested:

- LM Control - MAPOUT.
- Map Control.

Method:

The Cache to Map conflict situation is tested by executing a 2-micrand sequence. Both micrands always result in a Cache miss, but the Map hit/miss is controlled to create variations of the test. The first micrand is issued from Instruction Fetch (IF), the second from Instruction Issue (II).

There are three variations tested in this subsection.

1. The first micrand (IF fetch) results in a Map miss. A Page Table search must complete while the second request waits in LM. The second micrand (II read) has a Map hit.
2. Same as test 1 except both micrands result in a Map hit.
3. Same as test 1 except a delay is introduced between micrand 1 and micrand 2. This causes the second request to arrive at the very edge of not creating a conflict.

Sixteen passes are executed to complete this subsection. Each pass is identical. The multiple passes are used to sense for timing and circuit instability.

Test Patterns:

Refer to Conditions.

Micrands Used:

| Number | Description                                       |
|--------|---------------------------------------------------|
| E      | Write Word to SVA.                                |
| F      | Purge Cache All.                                  |
| 10     | Set P Descriptor register.                        |
| 2C     | Double Read using two SVAs; IF/II Requests.       |
| 2D     | Double Read using two SVAs; IF/delay/II Requests. |
| 12     | Purge Map All.                                    |

Initialization:

- Purge All Cache.
- Set P Descriptor register to 0--0.
  - Set RF11 to 0--0; Segment number.
  - Set CM 0 to 0--0; P Descriptor value.
  - Start processor at micrand 10, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition numbers to -1(FE(HEX)).
- Set CM; 0 through FF(HEX), to C3--C30000XXH, XX = 0 through FF(HEX).
- Set CM 1000(HEX) to F000 0000 0000 0000(HEX) (IF PTD).
- Set CM 1001(HEX) to B010 0000 0000 0001(HEX) (II PTD).
- Set pass count to 1.

Conditions:

NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.1, C.2 error, go to C.3.  
Pass count (1 through 10(HEX)).  
a. Report pass count; informational.
- C.1 Execute test 1, report data from IF read.  
a. Start processor at micrand 12, purge Map all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set RF2 to 0--0(HEX); IF SVA, Set up Map.  
d. Set RFA to C3--C3000000(HEX); write data.  
e. Start processor at micrand E, write word to SVA, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Start processor at micrand F, purge Cache All, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Set RF2 to 0100 0000 0000(HEX); II SVA.  
j. Set RF10 to 0--0; IF PVA.  
k. Set RF8 to DF--DF; filler - II Result register.  
l. Set RF14 to DF--DF; filler - IF Result register.  
m. Start processor at micrand 2C, double read from two SVAs; IF/II requests, wait for CPU halted.  
n. Read CPU status, format and save if error; error if not 0.  
o. Read RF14, report data; error if not A5--A5C3C3(HEX).
- C.2 Test 1, report data from II read.  
a. Read RF8, report data; error if not C3--C3000040(HEX).

If Scope Mode set, go to C.0.  
If error on C.1, C.2, go to C.7.

- C.3 If Scope Mode set and no C.3, C.4 error, go to C.5.  
Execute test 2, report data from IF read.  
a. Start processor at micrand F, purge Cache all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set RF8 to DF--DF; filler - II Result register.  
d. Set RF14 to DF--DF; filler - IF Result register.  
e. Start processor at micrand 2C, double read from two SVAs; IF/II requests, wait for CPU halted.  
f. Read CPU status, format and save if error; error if not 0.  
g. Read RF14, report data; error if not A5--A5C3C3(HEX).
- C.4 Test 2, report data from II read.  
a. Read RF8, report data; error if not C3--C30040(HEX).

If Scope Mode set, go to C.0.  
If error on C.3, C.4, go to C.7.

60000285 A

LMT3/P - 540

Conditions:

- C.5 Execute test 3, report data from IF read.  
a. Start processor at micrand 12, Purge Map All, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set RF2 to 0--0(HEX); IF SVA, Set up Map.  
d. Start processor at micrand E, write word to SVA, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Start processor at micrand F, Purge Cache All, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Set RF2 to 0100 0000 0000(HEX); II SVA.  
i. Set RF8 to DF--DF; filler - II Result register.  
j. Set RF14 to DF--DF; filler - IF Result register.  
k. Start processor at micrand 2D, double read from two SVAs; IF/Delay/II requests, wait for CPU halted.  
l. Read CPU status, format and save if error; error if not 0.  
m. Read RF14, report data; error if not A5--A5C3C3(HEX).
- C.6 Test 3, report data from II read.  
a. Read RF8, report data; error if not C3--C30040(HEX).

If Scope Mode set, go to C.0.

- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.
- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 10(HEX) passes complete; else return to C.0.

Section 48, Subsection 1

This subsection tests two unrelated conflicts.

1. Read access, or read/miss, read/hit conflict. Data from both requests try to use the RDATA trunk from LM to OPI at the same time.
2. Read/write, or block fill followed by a Cache Write conflict. Data coming from CM due to a block fill request and data from a write operation are requesting storage into Cache at the same time.

Hardware Tested:

- Read Access Fill Network.
- LM Control.

60000285 A

LMT3/P - 541

# Method:

Conditions 0 and 1 test the read access delay conflict. A 2-micrand sequence is used. The first micrand, an IF operation, results in a Map hit and a Cache miss. The second micrand, an II read operation, results in a Cache hit and is timed such that the data coming from CM as a result of the block fill from micrand one and the data from Cache from micrand two request the RDATA trunk to OPI at the same time. The Cache hit data should be delayed one major cycle. If both operations complete as expected, the test is considered successful.

Conditions 2, 3 and 4 test the block fill/Cache write conflict. A 2-micrand sequence is also used here. Micrand one, an IF operation, results in a Map hit and a Cache miss. Micrand two, an II write operation, results in a Cache hit. The second micrand is timed such that it requests to write Cache at the same time the first word of the block fill initiated by micrand one is requesting write access to Cache. The second micrands write request must wait for all words of the block (if contiguous) to be stored before it's write can be completed.

Conditions 5, 6 and 7 repeat the block fill/Cache write conflict test using slightly different timing. The second micrand is now timed to request Cache write when the last word from the block fill is requesting Cache storage. The IF micrand is tested for completion, the write data from the second micrand is checked, both in CM and Cache.

## Test Patterns:

Refer to Conditions.

## Micrands Used:

| Number | Description                             |
|--------|-----------------------------------------|
| D      | Read Word from SVA.                     |
| F      | Purge Cache All.                        |
| 12     | Purge Map All.                          |
| 2E     | Double word load; IF/Delay/II Requests. |
| 2F     | Word Fetch (IF), Word Store (II).       |
| 30     | Double word load; IF/Delay/II Requests. |

## Initialization:

- Set CM; 0 through FF(HEX) to C3--C3 00xx(HEX); xx = 0 through FF (HEX).
- Set CM 1000(HEX) to B000 0000 0000 0000(HEX); PTD.

# Conditions:

## NOTE

Disregard skipped conditions on error display.

- C.0 If Scope Mode set and no C.0, C.1 error, go to C.2.  
Read access delay test, report IF result.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 12, purge Map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 0--020(HEX); II SVA.
  - Start processor at micrand D, read word from SVA, wait for CPU halted; initialize Map and Cache.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF10 to 0--0(HEX); IF PVA.
  - Set RF8 to DF--DF(HEX); filler for II result.
  - Set RF14 to DF--DF(HEX); filler for IF result.
  - Start processor at micrand 2E, IF/II Read, wait for CPU halted, execute Read Access Delay Test.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not A5--A5C3C3(HEX).
- C.1 Read access delay test, report II result.
- Read RF8, report data; error if not C3--C30004(HEX).
- If Scope Mode set, go to C.0.  
If error on C.0, C.1, go to C.8.
- C.2 If Scope Mode set and no C.2, C.3, C.4, go to C.5.  
Block fill/Cache write test, report IF result.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 12, purge Map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand D, read word from SVA, wait for CPU halted; set up Map and Cache.
  - Read CPU status, format and save if error; error if not 0.
  - Set RFA to E7--E70004(HEX); II write data.
  - Set RF14 to DF--DF(HEX); filler for IF.
  - Start processor at micrand 2F; IF/II write, wait for CPU halted; execute Block Fill/Write Test.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not A5--A5C3C3(HEX).
- C.3 Block fill/Cache write test, report II CM data.
- Read CM 4), report data; error if not E7--E7 0004(HEX).

# Conditions:

- C.4 Block fill/Cache write test, report II Cache data.
- Select FAKECM mode.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted; fetch Cache data.
  - Read CPU status, format and save if error; error if not 0.
  - Deselect FAKECM mode.
  - Read RF6, report data; error if not E7--E7 0004(HEX).

If Scope Mode set, go to C.0.

If error on C.2, C.3, C.4, go to C.8.

- C.5 Block fill/Cache write test, report IF result.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand 12, purge Map all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand D, read word from SVA, wait for CPU halted; set up Map and Cache.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF14 to DF--DF(HEX); filler for IF.
  - Start processor at micrand 30; IF/II write, wait for CPU halted; execute block fill/write test.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF14, report data; error if not A5--A5C3C3(HEX).

- C.6 Block fill/Cache write test, report II CM data.
- Read CM 4), report data; error if not E7--E7 0004(HEX).

- C.7 Block fill/Cache write test, report II Cache data.
- Select FAKECM mode.
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted; fetch Cache data.
  - Read CPU status, format and save if error; error if not 0.
  - Deselect FAKECM mode.
  - Read RF6, report data; error if not E7--E7 0004(HEX).

If Scope Mode set, go to C.0.

- C.8 CPU Status Word 1.
- Report word 1 if saved; error if reported.

- C.9 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.10 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

60000285 A

LMT3/P - 544

## Section 48, Subsection 2

This subsection tests all situations where Cache allocation is requested but the block to be allocated is being filled and the corresponding blocks in other available sets are also unallocatable.

### Hardware Tested:

- Allocation Not Possible.
- Cache Set Allocatable.

### Method:

The sequence used to create this allocation conflict consists of two micrands. The first micrand, a II read, results in a Cache miss, Map hit condition. The second micrand is also a II read with its Cache block address (SVA bits 51 through 58) equal to the first micrands Cache block address, but the Tag portions are different. The second SVA also misses Cache and a second allocate to the same block is generated. Because the first micrand still has words coming from CM to fill its allocated block, the Word Valid bits are clear and the set is not allocatable. The second micrand is therefore treated as a Bypass Read and only the single word is fetched from CM and sent back to the Register File. There are three variations of this forced bypass read. Four passes are required to test all Cache sets, one set for each pass. Different variations are used during each pass as follows:

- |        |                                                                                                    |
|--------|----------------------------------------------------------------------------------------------------|
| Pass 1 | Both reads result in Page Map hits.                                                                |
| Pass 2 | The first read results in a Map miss, Page Table hit. The second read always results in a Map hit. |
| Pass 3 | The first read results in a Map miss and a Page Table update.                                      |
| Pass 4 | Same as pass 3 except test Set 3 if available; else test Set 1.                                    |

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description                                |
|--------|--------------------------------------------|
| D      | Read Word from SVA.                        |
| E      | Write Word to SVA.                         |
| F      | Purge Cache All.                           |
| 12     | Purge Map All.                             |
| 1A     | Double Read from two SVAs; II/II requests. |

60000285 A

LMT3/P - 545

#### Initialization:

- Set CM; 0 through FF(HEX) to C3--C30000xx(HEX); xx = 0 through FF (HEX).
- Set CM 1001(HEX) to B010 0000 0000 0001(HEX); PTD for first SVA.
- Set pass count to 1.

#### Conditions:

##### NOTE

Disregard skipped conditions on error display.

- C.0 Pass count (1 through 4).  
a. Report pass count; informational.
- C.1 Report current Cache Set being tested.  
a. If only 2 sets available, sequentially select Cache sets 0, 1, 0, 1 on successive passes; else select sets 0, 1, 2, 3.  
b. Report current Set.
- C.2 Execute test, report result of first read.  
a. Start processor at micrand F, purge Cache all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Start processor at micrand 12, purge Map all, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. If pass 1 or 2, set CM; 1000(HEX), to F000 0000 0000 0000(HEX); PTD for first SVA.  
f. If pass 3 or 4, set CM; 1000(HEX), to C000 0000 0000 0000(HEX); PTD for first SVA.  
g. Set RF2 to 0100 0000 0000(HEX); SVA for second read.  
h. Set RFA to C3--C3000040(HEX); write data.  
i. Start processor at micrand E, write word to SVA, wait for CPU halted; initialize Map.  
j. Read CPU status, format and save if error; error if not 0.  
k. Set RF2 to 0--0(HEX); SVA for first read.  
l. Set RFA to C3--C3000000; write data.  
m. If pass 1, Start processor at micrand E, write word to SVA, wait for CPU halted; initialize Map.  
n. If pass 1, Read CPU status, format and save if error; error if not 0.  
o. Set RF3 to 0100 0000 0000(HEX); SVA for second read.  
p. Set RF6 to DF--DF(HEX); filler for first read.  
q. Set RF7 to DF--DF(HEX); filler for second read.  
r. Start processor at micrand 1A, double read from two SVAs; 11/11 requests, wait for CPU halted.  
s. Read CPU status, format and save if error; error if not 0.  
t. Read RF6, report data; error if not C3--C3000000(HEX).
- C.3 Report result of second read.  
a. Read RF7, report data; error if not C3--C3000040(HEX).

80000285 A

LMT3/P - 546

#### Conditions:

Select FAKECM mode.

If Scope Mode set and no C.1, C.3, C.4 error, go to C.5.

- C.4 Report word 0 from Cache.  
a. Select FAKECM mode.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000000(HEX)

If Scope Mode set, go to C.0.

If error on C.2, C.3, C.4, go to C.9.

- C.5 If Scope Mode is set and no C.5 error, go to C.6.  
Report word 1 from Cache.  
a. Set RF2 to 0000 0000 0008(HEX); SVA for word 1.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000001(HEX).

If Scope Mode set, go to C.0.

If error on C.5, go to C.9.

- C.6 If Scope Mode set and no C.6 error, go to C.7.  
Report word 2 from Cache.  
a. Set RF2 to 0000 0000 0010(HEX); SVA for word 2.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000002(HEX).

If Scope Mode set, go to C.0.

If error on C.6, go to C.9.

- C.7 If Scope Mode set and no C.7 error, go to C.8.  
Report word 3 from Cache.  
a. Set RF2 to 0000 0000 0018(HEX); SVA for word 3.  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not C3--C3000003(HEX).

If Scope Mode set, go to C.0.

If error on C.7, go to C.9.

80000285 A

LMT3/P - 547

# Conditions:

- C.8 Report word 0 from Cache (second read).
- Set RF2 to 0100 0000 0000(HEX).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero.
  - Read RF8, report data; error if not 0.

If Scope Mode set go to C.0.

- C.9 CPU Status Word 1.
- Report word 1 if saved; error if reported.
- C.10 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.11 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM mode.

Exit Subsection if four passes done; else return to C.0.

## Section 48, Subsection 3

This subsection tests two situations that can occur during a Page Table Update which result in loss of the requestors RMA.

### Hardware Tested:

- Page Map Control.

### Method:

Two specific situations are tested in which the RMA retrieved from the Page Table is lost due to a conflict.

Conditions 0, 1, 2 and 3 test the situation in which no Page Map Sets are available and a requested operation requires a Page Table Update. If the Update is allowed to proceed, the RMA from the Page Table will be lost (no Map to hold it). Logic is used, therefore, to block the update until the RMA is sent to CMC. A write operation is used to test this case. The Page Table should be updated and the data in RFA should be written to CM 1.

Condition 4 tests the situation in which CMC becomes busy during a Page Table search, after a hit but before the last request has been sent to CMC. For example, if a hit occurred on the first Page Table entry (even) and the next odd request is held up due to CM refresh, the RMA derived from the even entry is lost.

### Test Patterns:

Refer to Conditions.

### Micrands Used:

| Number | Description         |
|--------|---------------------|
| D      | Read Word from SVA. |
| E      | Write Word to SVA.  |
| 12     | Purge Map All.      |

### Initialization:

- Select all Map Sets.
- Purge all Map Sets.
- Deselect all Cache Sets.
- Set CM; 0 through F(HEX), to C3--C300000x(HEX); x = 0 through F(HEX).
- Set CM 1000(HEX) to E000 0000 0000 0000(HEX).
- Set CM 1001(HEX) to 0--0.

### Conditions:

- C.0 Report Descriptor before update.
- Deselect all Page Map Sets.
  - Set RF2 to 0--08(HEX); SVA, word 1.
  - Set RFA to E7--E7(HEX); write data.
  - Report E0--0(HEX); informational.
- C.1 Execute test 1, report Descriptor after update.
- Set CM 1 to C3--C3000001(HEX); filler.
  - Start processor at micrand E, write RFA to memory (SVA), wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 1000(HEX), report data; error if not F0--0(HEX); modify bit set.
  - Set CM 1000(HEX) to E0--0(HEX), clear Modify bit for condition loop.
- C.2 Report data from CM 1, write data.
- Read CM 1, report data; error if not E7--E7(HEX).

Conditions:

- C.3 Report data from CM 0.  
a. Read CM 0, report data; error if not C3--C3000000(HEX).
- C.4 Select Maps and do nothing.  
a. Select all Map Sets.  
b. Report zeros and exit condition.
- If Scope Mode is set, go to C.0.
- C.5 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.
- C.6 CPU Status Word 2.  
a. Report Word 2 if saved; error if reported.
- C.7 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect Non-interleaved mode.

SECTION 49 - INVALIDATE CONFLICTS

This section tests Cache Invalidate operations when started and executed during previously tested LM conflict situations.

Subsection 0 - Invalidate Conflict, Write/Cache Hit.

Subsection 1 - Invalidate Conflict, Page Table Update.

Subsection 2 - Invalidate Conflict, Cache Lookahead.

Subsection 3 - Invalidate Conflict, Cache to Map.

Subsection 4 - Invalidate Conflict, Wait Allocate.

Subsection 5 - Invalidate Conflict, Cache Purge Block.

Method:

Cache Invalidate conflict testing is accomplished by starting a processor micrand sequence, then, not waiting for CPU Halted, executing a Cache Invalidate from the IOU. Because the IOU is much slower than the CPU a string of No-op micrands precedes the test micrands. The time between the Start processor and the start of the invalidate is constant. This is the time that must be padded with No-op micrands. On the first pass through the test conditions the timing is such that the Invalidate actually starts before the test sequence. On each successive pass the CPU sequence is started at a point where one fewer No-op micrands is executed. This allows the test micrands to start one major earlier, which effectively puts the Invalidate one major cycle further into the CPU test sequence. This process continues until the Invalidate is starting after the CPU test sequence has finished.

Example:

|                                   |                 |
|-----------------------------------|-----------------|
| Start pass 1 - through            | No-op micrand.  |
| Start pass 2 - through            | No-op micrand.  |
| Start pass 3 - through            | No-op micrand.  |
| Start pass 4 - through            | No-op micrand.  |
| Issue Invalidate pass 1 - through | No-op micrand.  |
| Issue Invalidate pass 2 - through | Test micrand 1. |
| Issue Invalidate pass 3 - through | Test micrand 2. |
| Issue Invalidate pass 4 - through | No-op micrand.  |

The number of passes executed in a specific subsection depends on the number of major cycles the test sequence takes. In the above example the test sequence takes two major cycles. Therefore, to start one major cycle before and end one major cycle after, four passes are required. To have a large enough window to allow for some timing differences, the Invalidate is started three major cycles before the test and allowed to continue for three major cycles past the end of the test sequence.

This method ensures the Invalidate occurs at every possible point during execution of the test sequence.



Test initialization and execution must be included in one condition to allow valid condition looping.

#### Initialization:

- Set PTA to 8.
- Set PTL to 0.
- Set PSM to 78(HEX).
- Select all Map Sets.
- Select all Available Cache Sets.

#### Section 49, Subsection 0

This subsection tests a simple write operation (Cache hit) when Cache Invalidate instructions are used to interrupt and delay the process.

#### Hardware Tested:

- Control.

#### Method:

Data is set into Cache for the invalidate sequence and also to ensure the write operation a Cache hit. The last No-op micrand in the No-op string is modified to jump to the test sequence; for example, the No-op string is modified to jump to micrand E. Micrand E will execute and halt. The processor is started, the invalidate is initiated and CPU halted is sensed for. After the CPU has halted both the write and the Invalidate operations are checked to ensure they completed normally.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                             |
|--------|-----------------------------------------|
| E      | Write Word to SVA.                      |
| D      | Read Word to SVA.                       |
| F      | Purge Cache All.                        |
| 12     | Purge Map All.                          |
| A0     | No-op Micrand string - jump to micrand. |

#### Initialization:

- Purge Cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Purge Map all.
  - Start processor at micrand 12, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set CM; 0 through FFF(HEX), to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Set CM 1000(HEX) to FFFF F1FF FF00 0000(HEX); PTD for invalidate data.
- Set CM 1001(HEX) to B000 0000 0000 0000(HEX); PTD for write data.
- Modify last micrand in A0 string to jump to micrand E.
- Set pass count to 1.

#### Conditions:

##### NOTES

Disregard skipped conditions on error display.

Skip this subsection if running on CPU-1.

- C.0 Initialize Cache data, report pass count (1 through 7).
- Set RF2 to FFFF 0FFF F800(HEX); SVA for Invalidate.
  - Start processor at micrand D, read word from SVA, wait or CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 0--0(HEX) (SVA).
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Report pass count; informational.
- C.1 Execute test, report write data from CM.
- Set CM 100(HEX) to C3--C3000100(HEX).
  - Set RFA to E7--E7 0000(HEX); write data.
  - Start processor at micrand A0 + pass count.
  - Invalidate CM address 100(HEX).
  - Wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM 0, report data; error if not E7--E7 0000(HEX).

Select FAKECM mode.

#### Conditions:

- C.2 If Scope Mode set and no C.0, C.1, C.2 error, go to C.3.  
Report write data from Cache.  
a. Set RF6 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word rom SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF6, report data; error if not E7--E7 0000(HEX).

If Scope Mode set, go to C.0.  
If error on C.0, C.1, C.2, go to C.4.

- C.3 Report data from Invalidated Cache word.  
a. Set RF2 to FFFF 0FFF F800(HEX) (SVA).  
b. Set RF6 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF6, report data; error if not 0.

If Scope Mode set, go to C.0.

- C.4 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

- C.5 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

- C.6 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if seven passes done; else return to C.0.

#### Section 49, Subsection 1

This subsection tests a Page Table Update operation when interrupted by a Cache Invalidate instruction. Ensure both operations complete normally.

#### Hardware Tested:

- LM Control.
- Map Update Control.

#### Method:

A Page Table Descriptor is set into CM at address 1001(HEX). The PTD has only the Valid bit set. When a write CM operation (Micrand E) is issued and a PTD match occurs, the PTD must be updated, setting its Modify bit. During this process a Cache Invalidate is issued causing the Map Update to be interrupted and delayed.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                                         |
|--------|-----------------------------------------------------|
| D      | Read Word from SVA.                                 |
| E      | Write Word to SVA.                                  |
| F      | Purge Cache All.                                    |
| 12     | Purge Map All.                                      |
| A0     | No-op micrand string for Invalidate conflict tests. |

#### Initialization:

- Set CM; 0 through F(HEX) to C3--C300000x(HEX); x = 0 through F(HEX).
- Set pass count to 1.

#### Conditions:

#### NOTES

Disregard skipped conditions on error display.

Skip this subsection if running on CPU-1.

- C.0 Pass count (1 through 14(HEX).  
a. Report pass count; informational.
- C.1 Report PTD entry before update.  
a. Report A000 0000 0000 0000(HEX); informational.

# Conditions:

- C.2 Execute test, report PTD after update.
- Set CM 1001(HEX) to A000 0000 0000 0000(HEX); PTD to update.
  - Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to FFFF 0FFF F800(HEX); SVA for Invalidate.
  - Start processor at micrand D, read word from SVA; initialize data to Invalidate, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to 0--0(HEX); SVA for write.
  - Start processor at micrand D, read word from SVA; initialize Cache for write data, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Start processor at micrand I2, Purge Map All, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set CM 100(HEX) to C3--C3000100(HEX).
  - Set RFA to E7--E7 0000(HEX); write data.
  - Start processor at micrand A0 + Pass Count.
  - Invalidate CM address 100(HEX).
  - Wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read CM; 1001(HEX), report data; error if not B000 0000 0000 0000 (HEX).

- C.3 Report data written to CM.
- Read CM 0, report data; error if not E7--E7 0000(HEX).

Select FAKECM mode.

- C.4 If Scope Mode set and no C.2, C.3, C.4 error, go to C.5.  
Report data written to Cache.
- Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not E7--E7 0000(HEX).

If Scope Mode set, go to C.0.  
If error on C.2, C.3, C.4, go to C.6.

- C.5 Report data from Invalidate Cache word.
- Set RF2 to FFFF 0FFF F800(HEX) (SVA).
  - Set RF8 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF8, report data; error if not 0.

If Scope Mode set, go to C.0.

- C.6 CPU Status Word 1.
- Report word 1 if saved; error if reported.

80000285 A

LMT3/P - 556

# Conditions:

- C.7 CPU Status Word 2.
- Report word 2 if saved; error if reported.
- C.8 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 14(HEX) passes done; else return to C.0.

## Section 49, Subsection 2

This subsection tests the Unconditional Lookahead operation when interrupted by a Cache Invalidate instruction. The tests ensures that both operations complete normally.

## Hardware Tested:

- Lookahead Control.
- Start Block Control.

## Method:

A Lookahead operation is initiated by issuing a Cache hit read. Therefore, a Cache block is filled using a read operation followed by a second read to the same block. The second read causes a Lookahead to be requested. During this process a Cache Invalidate is issued causing the Lookahead to be interrupted and delayed.

## Test Patterns:

Refer to Conditions.

## Micrands Used:

| Number | Description                                           |
|--------|-------------------------------------------------------|
| D      | Read Word from SVA.                                   |
| F      | Purge Cache all.                                      |
| A0     | No-op micrand string for Invalidate conflict testing. |

80000285 A

LMT3/P - 557

#### Initialization:

- Set pass count to 1.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Modify last micrand in A0 string to jump to micrand D.
- Select Unconditional Lookahead.

#### Conditions:

#### NOTES

Disregard skipped conditions on error display.

Skip this subsection if running on CPU-1.

- C.0 Pass count (1 through 16(HEX)).  
a. Report pass count; informational.
- C.1 Execute test, report data from second read.  
a. Start processor at micrand F, purge Cache all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Set RF2 to FFFF 0FFF F800(HEX); SVA for Invalidate.  
d. Start processor at micrand D, read word from SVA; initialize data to Invalidate, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Set RF2 to 0--0(HEX); SVA for read.  
g. Start processor at micrand D, read word from SVA; initialize Cache for Lookahead, wait for CPU halted.  
h. Read CPU status, format and save if error; error if not 0.  
i. Set CM 100(HEX) to C3--C3000100(HEX).  
j. Set RF8 to DF--DF(HEX); filler.  
k. Start processor at micrand A0 + pass count.  
l. Invalidate CM address 100(HEX).  
m. Wait for CPU halted.  
n. Read CPU status, format and save if error; error if not 0.  
o. Read RF8, report data; error if not C3--C3000000(HEX).

#### Select FAKECM mode.

- C.2 If Scope Mode set and no C.1, C.2 error, go to C.3.  
Report Cache word 0 from Lookahead block.  
a. Set RF2 to 0--020(HEX); SVA, word 0.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C3000004(HEX).

If Scope Mode set, go to C.0.  
If error on C.1, C.2, go to C.7.

60000285 A

LMT3/P - 558

#### Conditions:

- C.3 If Scope Mode set and no C.3 error, go to C.4.  
Report Cache word 1 from Lookahead block.  
a. Set RF2 to 0--028(HEX); SVA, word 1.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C3000005(HEX).

If Scope Mode set, go to C.0.  
If error on C.3, go to C.7.

- C.4 If Scope Mode set and no C.4 error, go to C.5.  
Report Cache word 2 from Lookahead block.  
a. Set RF2 to 0--030(HEX); SVA, word 2.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C3000006(HEX).

If Scope Mode set, go to C.0.  
If error on C.4, go to C.7.

- C.5 If Scope Mode set and no C.5 error, go to C.6.  
Report Cache word 3 from Lookahead block.  
a. Set RF2 to 0--038(HEX); SVA, word 3.  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not C3--C3 0007(HEX).

If Scope Mode set, go to C.0.  
If error on C.5, go to C.7.

- C.6 Report data from invalidated Cache word.  
a. Set RF2 to FFFF 0FFF F800(HEX) (SVA).  
b. Set RF8 to DF--DF(HEX); filler.  
c. Start processor at micrand D, read word from SVA, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Read RF8, report data; error if not 0.

If Scope Mode set, go to C.0.

- C.7 CPU Status Word 1.  
a. Report word 1 if saved; error if reported.

- C.8 CPU Status Word 2.  
a. Report word 2 if saved; error if reported.

- C.9 First Failure Capture Information.  
a. Report PFS87 bits 19-23 in bits 59-63; information only.  
b. Report last PTL written in bits 34-47; information only.  
c. Report last PTA written in bits 13-31; information only.

60000285 A

LMT3/P - 559

#### Conditions:

Deselect FAKECM Mode.

Exit Subsection if 16(HEX) passes complete; else return to C.0.

Deselect Lookahead.

#### Section 49, Subsection 3

This subsection tests the Cache-to-Map conflict situation when interrupted by Cache Invalidate instructions. Ensure all micrands complete normally.

#### Hardware Tested:

- LM Control.
- Map Control.

#### Method:

A Cache-to-Map conflict occurs when two Cache miss read operations, issued from different sources, arrive at LM such that the second read must wait for the first to finish with the Page Map.

The Cache-to-Map conflict in this subsection is created by issuing an Instruction Fetch (IF) which causes a Cache and Map miss. The second micrand arrives during Map update from Instruction Issue (II) and must wait for the Map to become free.

During this process a Cache Invalidate is issued causing the operation to be interrupted and delayed.

#### Test Patterns:

Refer to Conditions.

#### Micrands Used:

| Number | Description                                           |
|--------|-------------------------------------------------------|
| D      | Read Word from SVA.                                   |
| F      | Purge Cache All.                                      |
| 12     | Purge Map All.                                        |
| 10     | Set P Descriptor register.                            |
| 2C     | Double Read using Two SVAs; IF/II requests.           |
| A0     | No-op micrand string for Invalidate conflict testing. |

60000285 A

LMT3/P - 580

#### Initialization:

- Set pass count to 1.
- Set CM; 0 through FFF(HEX) to C3--C3000xxx(HEX); xxx = 0 through FFF(HEX).
- Set P Descriptor register to zero.
  - Set RF11 to 0--0; segment number.
  - Set CM 0 to 0--0; P Descriptor value.
  - Start processor at micrand 10, set P Descriptor register, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Modify last micrand in A0 string to jump to micrand 2C.

#### Conditions:

#### NOTE

Skip this subsection if running on CPU 1.

- C.0 Pass count (1 through 15(HEX).  
a. Report pass count; informational.
- C.1 Execute test, report data from IF read.  
a. Start processor at micrand F, purge Cache all, wait for CPU halted.  
b. Read CPU status, format and save if error; error if not 0.  
c. Start processor at micrand 12, purge Map all, wait for CPU halted.  
d. Read CPU status, format and save if error; error if not 0.  
e. Set RF2 to FFFF 0FFF F800(HEX); SVA for Invalidate.  
f. Start processor at micrand D, read word from SVA; initialize Invalidate data, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Set CM 100(HEX) to C3--C3000100(HEX).  
i. Set RF10 to 0--0; PVA for IF.  
j. Set RF2 to 0--020(HEX); SVA for II.  
k. Set RF14 to DF---DF(HEX); filler for IF read.  
l. Set RF8 to DF---DF(HEX); filler for II read.  
m. Start processor at micrand A0 + pass count.  
n. Invalidate CM address 100(HEX).  
o. Wait for CPU halted.  
p. Read CPU status, format and save if error; error if not 0.  
q. Read RF14, report data; error if not A5--A5C3C3(HEX).
- C.2 Report data from II read.  
a. Read RF8, report data; error if not C3--C3 0004(HEX).

60000285 A

LMT3/P - 581

#### Conditions:

- C.3 Report data from Invalidated Cache word.
- Select FAKECM mode.
  - Set RF2 to FFFF 0FFF F800(HEX) (SVA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0.

- C.4 CPU Status Word 1.
- Report word 1 if saved; error if reported.

If Scope Mode set, go to C.1.

- C.5 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.6 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 15(HEX) passes complete; else return to C.0.

#### Section 49, Subsection 4

This subsection tests the Wait Allocate conflict situation when interrupted by Cache Invalidate instructions. Ensure all micrands complete normally.

#### Hardware Tested:

- LM Control.
- Wait Allocate Network.

#### Method:

Wait Allocate occurs when a Cache miss read requests Cache allocation during the same minor cycle a data word is received from CM due to a previous Cache miss read (block fill).

This process is initiated, then interrupted and delayed by Cache Invalidate instructions.

#### Test Patterns:

Refer to Conditions.

60000285 A

LMT3/P - 562

#### Micrands Used:

| Number | Description                                           |
|--------|-------------------------------------------------------|
| D      | Read word from SVA.                                   |
| F      | Purge Cache all.                                      |
| 1A     | Double read using two SVAs; 11/11 requests.           |
| A0     | No-op micrand string for Invalidate conflict testing. |

#### Initialization:

- Set pass count to 1.
- Modify last micrand in A0 string to jump to micrand 1A.
- Initialize Map.
  - Set RF2 to 0--0(HEX) (SVA).
- Start processor at micrand D, read word from SVA, wait for CPU halted.
- Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(HEX).

#### Conditions:

#### NOTE

Skip this subsection if running on CPU-1.

- C.0 Pass count (1 through 1C(HEX).
- Report pass count; informational.
- C.1 Execute test, report data from first read.
- Start processor at micrand F, purge Cache all, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set RF2 to FFFF 0FFF F800(HEX); SVA for Invalidate.
  - Start processor at micrand D, read word from SVA; initialize Invalidate data, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Set CM; 100(HEX) to C3--C3000100(HEX).
  - Set RF2 to 0--0(HEX); SVA for first read.
  - Set RF3 to 0--020(HEX); SVA for second read.
  - Set RF6 to DF--DF(HEX); filler for first read.
  - Set RF7 to DF--DF(HEX); filler for second read.
  - Start processor at micrand A0 + pass count.
  - Invalidate CM address 100(HEX).
  - Wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not C3--C3000000(HEX).
- C.2 Report data from second read.
- Read RF7, report data; error if not C3--C3000004(HEX).

60000285 A

LMT3/P - 563

Conditions:

- C.3 Report data from Invalidated Cache word.
- Select FAKECM mode.
  - Set RF2 to FFFF 0FFF F800(HEX) (SVA).
  - Set RF6 to DF--DF(HEX); filler.
  - Start processor at micrand D, read word from SVA, wait for CPU halted.
  - Read CPU status, format and save if error; error if not 0.
  - Read RF6, report data; error if not 0.

If Scope Mode set, go to C.1.

- C.4 CPU Status Word 1.
- Report word 1 if saved; error if reported.

- C.5 CPU Status Word 2.
- Report word 2 if saved; error if reported.

- C.6 First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Deselect FAKECM Mode.

Exit Subsection if 1C(HEX) passes complete; else return to C.0.

#### Section 49, Subsection 5

This subsection tests a Cache Purge operation when interrupted by a Cache Invalidate instruction. The test ensures that both operations complete normally.

Hardware Tested:

- LM Control.

Method:

The Purge Cache Block operation is used because it involves a virtual instruction fetch and because it requires the least amount of time of all the Cache Purge operations.

All Cache is initialized (word 0 only) and the last Cache block (512 bytes) is purged. During this process a Cache Invalidate is issued causing the Block Purge to be interrupted and delayed.

Conditions 0 and 1 are used to initialize all Cache except the last 512 bytes. FAKECM is used to allocate and write a formatted pattern to word 0 of each block. These two conditions are repeated 240 times.

60000285 A

LMT3/P - 564

Conditions 2 through 9 comprise the test loop. Condition 2 initiates a block read, using the Invalidate SVA, initializing the Cache block to be Invalidated.

Conditions 3 and 4 are used to reinitialize the Cache locations (16 four-word blocks) purged during the block purge.

Condition 5 executes the purge and invalidate test, then reads Cache at the invalidated address to ensure the invalidate completed successfully.

Conditions 6 and 7 read all Cache locations, word 0 only, and verify that the purge completed correctly.

The operation is then repeated until all passes have been completed or until an error is sensed.

Test Patterns:

Data used to initialize word 0 of all Cache locations is formatted as: E7E7 xxxx xxxx(HEX).

x-x = Current SVA.

The purge SVA is 0--01FE0(HEX). This should purge Cache addresses 0--01E00(HEX) through 0--01FE0(HEX).

Cache addresses 0--0 through 0--01DE0(HEX) should never be purged.

Micrands Used:

| Number | Description                                         |
|--------|-----------------------------------------------------|
| C      | FAKECM allocate Cache.                              |
| D      | Read word from SVA.                                 |
| E      | Write word to SVA.                                  |
| F      | Purge Cache all.                                    |
| 13     | Enter P register; start IF.                         |
| 28     | Virtual purge micrand.                              |
| A0     | No-op micrand string for Invalidate conflict tests. |

Initialization:

- Set pass count to 1.
- Select all available Cache Sets.
- Purge Cache all.
  - Start processor at micrand F, wait for CPU halted.
  - Read CPU status, format and save if error; error if not zero. If error, set failing condition number to -1(FE(HEX)).
- Set Purge Micrand (Micrand 28) into Control Store at address 105(HEX).
- Select FAKECM mode.
- Set current SVA to zero.
- Modify last micrand in A0 string to jump to Micrand 13.

60000285 A

LMT3/P - 565

Conditions:

NOTES

Disregard skipped conditions on error display.

Skip this subsection if running on CPU-1.

- C.0 If Scope Mode is set and  $I < F0(\text{HEX})$ , execute C.0 and C.1.  
Allocate Cache, report current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current SVA; informational.

- C.1 Write Cache, report data.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write data; informational.

If error on C.0, C.1, go to C.8.

Continue to next condition if  $F0(\text{HEX})$  SVA's have been written; else advance SVA (+20(HEX)) and return to C.0.

If Scope Mode is set and  $F1(\text{HEX}) < I < 100(\text{HEX})$ , execute C.2, C.3, C.4.

Conditions:

- C.2 Initialize Cache Invalidate block, report pass count (1 through 20 (HEX))  
a. Set RF10 to 0--010000(HEX); P register value for virtual instruction.  
b. Set RF2 to 0--010000(HEX); SVA for Cache allocate.  
c. Set RFA to 050--0(HEX); Virtual Purge instruction, to be written to Cache.  
d. Start processor at Micrand C, FAKECM Allocate Cache, wait for CPU halted.  
e. Read CPU status, format and save if error; error if not 0.  
f. Start processor at micrand E, write data from RFA to Cache, wait for CPU halted.  
g. Read CPU status, format and save if error; error if not 0.  
h. Set RF2 to FFFF 0FFF F800(HEX); SVA for invalidate.  
i. Deselect FAKECM mode.  
j. Start processor at micrand D, read word from SVA, wait for CPU halted.  
k. Read CPU status, format and save if error; error if not 0.  
l. Select FAKECM mode.  
m. Report pass count; informational.

80000285 A

LMT3/P - 566

Conditions:

- C.3 Reallocate purged block, report current SVA.  
a. Set RF2 to current SVA.  
b. Start processor at micrand C, FAKECM allocate Cache, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report CURRENT SVA; informational.
- C.4 Write purge of block, report data.  
a. Set RFA to current write data.  
b. Start processor at micrand E, write word to SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Report current write data; informational.

If error on C.2, C.3, C.4, go to C.8.

Continue to next condition if error or if 10(HEX) SVAs have been written; else advance SVA (+20(HEX)) and return to C.3.

Set current SVA to 0.

- C.5 If Scope Mode is set and  $101(\text{HEX}) < I < 200(\text{HEX})$ , execute C.5, C.6, C.7. Issue Purge and Invalidate, report Cache invalidated data.  
a. Set CM 0 to 050--0(HEX); virtual purge instruction.  
b. Set RF10 to 0--0(HEX); P register value.  
c. Set RF2 to 0--01FE0(HEX); purge SVA.  
d. Deselect FAKECM mode.  
e. Set CM; 100(HEX) to C3--C30100(HEX).  
f. Start processor at micrand A0 + Pass Count.  
g. Invalidate CM address 100(HEX).  
h. Wait for CPU halted.  
i. Read CPU status, format and save if error; error if not 0.  
j. Select FAKECM mode.  
k. Set RF2 to FFFF 0FFF F800(HEX); invalidated SVA.  
l. Set RF8 to DF--DF(HEX); filler.  
m. Start processor at micrand D, read word from SVA, wait for CPU halted.  
n. Read CPU status, format and save if error; error if not 0.  
o. Read RF8, report data; error if not 0.

- C.8 Current read SVA.  
a. Set RF2 to current SVA.  
b. Report current SVA; informational.

- C.7 Read data from Cache, report data.  
a. Set RF8 to DF--DF(HEX); filler.  
b. Start processor at micrand D, read word from SVA, wait for CPU halted.  
c. Read CPU status, format and save if error; error if not 0.  
d. Read RF8, report data; error if not as written in blocks not purged 0--EF(HEX) or not 0 in purged blocks  $F0(\text{HEX})$ --FF(HEX).

80000285 A

LMT3/P - 567



Conditions:

Continue to next condition if error or if 100(HEX) SVAs have been read; else advance SVA (+20(HEX) and return to C.6.

If Scope Mode set, go to C.0.

Set current SVA to 1E00(HEX).

- C.8 CPU Status Word 1.
  - a. Report Word 1 if saved; error if reported.
- C.9 CPU Status Word 2.
  - a. Report Word 2 if saved; error if reported.
- C.10 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

Exit Subsection if 20(HEX) passes complete; else return to C.2.

Deselect FAKECM mode.

Reset Control Store Address 105(HEX) to original micrand (micrand 39).

SECTION 50 - TEST FUNCTION CODE BITS

This section tests function code bits using Central Memory Exchange Function.

Subsection 0 - Test Function Code Bits.

Method:

Refer to subsection method.

Initialization:

- None.

Section 50, Subsection 0

This subsection tests Local Memory function code bits using CM Exchange function, which issues a read and a write at the same CM address. Data read from and written into CM is checked.

Micrands Used:

| Number | Description      |
|--------|------------------|
| F      | Purge Cache all. |
| 3E     | CM exchange.     |

Initialization:

- Purge Cache all.
- Start processor at micrand F. Wait for CPU halted.
- Clear CM 0-FFF(HEX).

Conditions:

- C.0 Set up data.
  - a. Set RF0 to 0, CM 0.
  - b. Set RF8 to DF--DF(HEX), filler.
  - c. Write CM 0 to 0, data to be read to RF8.
  - d. Set RF7 to F--F(HEX), data to be written into CM 0.
- C.1 Execute test.
  - a. Start processor at micrand 3E, CM-exchange. Wait for CPU halted.
  - b. Read and format CPU status; error if not 0.
  - c. Read RF8, report data; error if not 0.

Conditions:

- C.2 Report data written in CM 0.
  - a. Read CM 0, report data; error if not F--F(HEX).
- C.3 Status word 1.
  - a. Report status word 1 if saved; error if reported.
- C.4 Status word 2.
  - a. Report status word 2 if saved; error if reported.
- C.5 M3 Register A4.
  - a. Report contents of A4 register; error if not 0.
- C.6 First Failure Capture Information.
  - a. Report PFS87 bits 19-23 in bits 59-63; information only.
  - b. Report last PTL written in bits 34-47; information only.
  - c. Report last PTA written in bits 13-31; information only.

SECTION 51 - TEST MAP HIT SIGNAL

This section does four tasks:

1. Initialize three Page Tables.
2. Do random read/write allocate.
3. Check final state of Page Table and Central Memory.
4. Check final state of page map by destroying Page Table and reading one word from each expected page descriptor in page map.

Subsection 0 - Part 1 and 2 above.

Subsection 1 - Part 3 and 4 above.

Hardware Tested:

- +MHIT.

Isolation Note:

This section is written to test the +MHIT signal. If this signal is bad, condition 2 should fail.

Test Patterns (hexadecimal):

a) PTA

| Register File Address | Data |
|-----------------------|------|
| RF18                  | 0    |
| RF19                  | 8    |
| I                     | I    |
| V                     | V    |
| RF1F                  | 38   |

b) Pattern for Subsection 0, Condition 1

| CM Address | Data |
|------------|------|
| CM 100     | 0    |
| CM 101     | 1    |
| I          | I    |
| V          | V    |
| CM 11F     | 1F   |

## c) Page Table 1

| Descriptor          | SVA                 |
|---------------------|---------------------|
| C002,0000,0800,0004 | 0000,0020,0000,4000 |
| C002,1000,0840,0004 | 0000,0021,0000,4208 |
| C002,2000,0880,0004 | 0000,0022,0000,4410 |
| C002,3000,08C0,0004 | 0000,0023,0000,4618 |
| C002,4000,0900,0004 | 0000,0024,0000,4820 |
| C002,5000,0940,0004 | 0000,0025,0000,4A28 |
| C002,6000,0980,0004 | 0000,0026,0000,4C30 |
| C002,7000,09C0,0004 | 0000,0027,0000,4E38 |
| C002,8000,0A00,0004 | 0000,0028,0000,5040 |
| C002,9000,0A40,0004 | 0000,0029,0000,5248 |
| C002,A000,0A80,0004 | 0000,002A,0000,5450 |
| C002,B000,0AC0,0004 | 0000,002B,0000,5658 |
| C002,C000,0B00,0004 | 0000,002C,0000,5860 |
| C002,D000,0B40,0004 | 0000,002D,0000,5A68 |
| C002,E000,0B80,0004 | 0000,002E,0000,5C70 |
| C002,F000,0BC0,0004 | 0000,002F,0000,5E78 |
| C003,0000,0C00,0004 | 0000,0030,0000,6080 |
| C003,1000,0C40,0004 | 0000,0031,0000,6288 |
| C003,2000,0C80,0004 | 0000,0032,0000,6490 |
| C003,3000,0CC0,0004 | 0000,0033,0000,6698 |
| C003,4000,0D00,0004 | 0000,0034,0000,68A0 |
| C003,5000,0D40,0004 | 0000,0035,0000,6AA8 |
| C003,6000,0D80,0004 | 0000,0036,0000,6CB0 |
| C003,7000,0DC0,0004 | 0000,0037,0000,6EB8 |
| C003,8000,0E00,0004 | 0000,0038,0000,70C0 |
| C003,9000,0E40,0004 | 0000,0039,0000,72C8 |
| C003,A000,0E80,0004 | 0000,003A,0000,74D0 |
| C003,B000,0EC0,0004 | 0000,003B,0000,76D8 |
| C003,C000,0F00,0004 | 0000,003C,0000,78E0 |
| C003,D000,0F40,0004 | 0000,003D,0000,7AE8 |
| C003,E000,0F80,0004 | 0000,003E,0000,7CF0 |
| C003,F000,0FC0,0004 | 0000,003F,0000,7EF8 |

## d) Page Table 2

| Descriptor          | SVA                 |
|---------------------|---------------------|
| C004,0000,1000,0004 | 0000,0040,0000,8000 |
| C004,1000,1040,0004 | 0000,0041,0000,8208 |
| C004,2000,1080,0004 | 0000,0042,0000,8410 |
| C004,3000,10C0,0004 | 0000,0043,0000,8618 |
| C004,4000,1100,0004 | 0000,0044,0000,8820 |
| C004,5000,1140,0004 | 0000,0045,0000,8A28 |
| C004,6000,1180,0004 | 0000,0046,0000,8C30 |
| C004,7000,11C0,0004 | 0000,0047,0000,8E38 |
| C004,8000,1200,0004 | 0000,0048,0000,9040 |
| C004,9000,1240,0004 | 0000,0049,0000,9248 |
| C004,A000,1280,0004 | 0000,004A,0000,9450 |
| C004,B000,12C0,0004 | 0000,004B,0000,9658 |
| C004,C000,1300,0004 | 0000,004C,0000,9860 |
| C004,D000,1340,0004 | 0000,004D,0000,9A68 |

80000285 A

LMT3/P - 572

## Page Table 2 (continued)

| Descriptor          | SVA                 |
|---------------------|---------------------|
| C004,E000,1380,0004 | 0000,004E,0000,9C70 |
| C004,F000,13C0,0004 | 0000,004F,0000,9E78 |
| C005,0000,1400,0004 | 0000,0050,0000,A080 |
| C005,1000,1440,0004 | 0000,0051,0000,A288 |
| C005,2000,1480,0004 | 0000,0052,0000,A480 |
| C005,3000,14C0,0004 | 0000,0053,0000,A688 |
| C005,4000,1500,0004 | 0000,0054,0000,A8A0 |
| C005,5000,1540,0004 | 0000,0055,0000,AAAB |
| C005,6000,1580,0004 | 0000,0056,0000,ACB0 |
| C005,7000,15C0,0004 | 0000,0057,0000,AEB8 |
| C005,8000,1600,0004 | 0000,0058,0000,B0C0 |
| C005,9000,1640,0004 | 0000,0059,0000,B2C8 |
| C005,A000,1680,0004 | 0000,005A,0000,B4D0 |
| C005,B000,16C0,0004 | 0000,005B,0000,B6D8 |
| C005,C000,1700,0004 | 0000,005C,0000,B8E0 |
| C005,D000,1740,0004 | 0000,005D,0000,BAE8 |
| C005,E000,1780,0004 | 0000,005E,0000,BCF0 |
| C005,F000,17C0,0004 | 0000,005F,0000,BEF8 |

## e) Page Table 3

| Descriptor          | SVA                 |
|---------------------|---------------------|
| C008,0000,2000,0004 | 0000,0080,0001,0000 |
| C008,1000,2040,0004 | 0000,0081,0001,0208 |
| C008,2000,2080,0004 | 0000,0082,0001,0410 |
| C008,3000,20C0,0004 | 0000,0083,0001,0618 |
| C008,4000,2100,0004 | 0000,0084,0001,0820 |
| C008,5000,2140,0004 | 0000,0085,0001,0A28 |
| C008,6000,2180,0004 | 0000,0086,0001,0C30 |
| C008,7000,21C0,0004 | 0000,0087,0001,0E38 |
| C008,8000,2200,0004 | 0000,0088,0001,1040 |
| C008,9000,2240,0004 | 0000,0089,0001,1248 |
| C008,A000,2280,0004 | 0000,008A,0001,1450 |
| C008,B000,22C0,0004 | 0000,008B,0001,1658 |
| C008,C000,2300,0004 | 0000,008C,0001,1860 |
| C008,D000,2340,0004 | 0000,008D,0001,1A68 |
| C008,E000,2380,0004 | 0000,008E,0001,1C70 |
| C008,F000,23C0,0004 | 0000,008F,0001,1E78 |
| C009,0000,2400,0004 | 0000,0090,0001,2080 |
| C009,1000,2440,0004 | 0000,0091,0001,2288 |
| C009,2000,2480,0004 | 0000,0092,0001,2490 |
| C009,3000,24C0,0004 | 0000,0093,0001,2698 |
| C009,4000,2500,0004 | 0000,0094,0001,28A0 |
| C009,5000,2540,0004 | 0000,0095,0001,2AA8 |
| C009,6000,2580,0004 | 0000,0096,0001,2CB0 |
| C009,7000,25C0,0004 | 0000,0097,0001,2EB8 |
| C009,8000,2600,0004 | 0000,0098,0001,30C0 |
| C009,9000,2640,0004 | 0000,0099,0001,32C8 |
| C009,A000,2680,0004 | 0000,009A,0001,34D0 |
| C009,B000,26C0,0004 | 0000,009B,0001,36D8 |
| C009,C000,2700,0004 | 0000,009C,0001,38E0 |
| C009,D000,2740,0004 | 0000,009D,0001,3AE8 |
| C009,E000,2780,0004 | 0000,009E,0001,3CF0 |
| C009,F000,27C0,0004 | 0000,009F,0001,3EF8 |

80000285 A

LMT3/P - 573

# f) Modified Page Table

| Descriptor          | SVA                 |
|---------------------|---------------------|
| F008,0000,2000,0004 | 0000,0100,0002,0000 |
| F008,1000,2040,0004 | 0000,0101,0002,0208 |
| F008,2000,2080,0004 | 0000,0102,0002,0410 |
| F008,3000,20C0,0004 | 0000,0103,0002,0618 |
| F008,4000,2100,0004 | 0000,0104,0002,0820 |
| F008,5000,2140,0004 | 0000,0105,0002,0A28 |
| F008,6000,2180,0004 | 0000,0106,0002,0C30 |
| F008,7000,21C0,0004 | 0000,0107,0002,0E38 |
| F008,8000,2200,0004 | 0000,0108,0002,1040 |
| F008,9000,2240,0004 | 0000,0109,0002,1248 |
| F008,A000,2280,0004 | 0000,010A,0002,1450 |
| F008,B000,22C0,0004 | 0000,010B,0002,1658 |
| F008,C000,2300,0004 | 0000,010C,0002,1860 |
| F008,D000,2340,0004 | 0000,010D,0002,1A68 |
| F008,E000,2380,0004 | 0000,010E,0002,1C70 |
| F008,F000,23C0,0004 | 0000,010F,0002,1E78 |
| F009,0000,2400,0004 | 0000,0110,0002,2080 |
| F009,1000,2440,0004 | 0000,0111,0002,2288 |
| F009,2000,2480,0004 | 0000,0112,0002,2490 |
| F009,3000,24C0,0004 | 0000,0113,0002,2698 |
| F009,4000,2500,0004 | 0000,0114,0002,28A0 |
| F009,5000,2540,0004 | 0000,0115,0002,2AA8 |
| F009,6000,2580,0004 | 0000,0116,0002,2CB0 |
| F009,7000,25C0,0004 | 0000,0117,0002,2EB8 |
| F009,8000,2600,0004 | 0000,0118,0002,30C0 |
| F009,9000,2640,0004 | 0000,0119,0002,32C8 |
| F009,A000,2680,0004 | 0000,011A,0002,34D0 |
| F009,B000,26C0,0004 | 0000,011B,0002,36D8 |
| F009,C000,2700,0004 | 0000,011C,0002,38E0 |
| F009,D000,2740,0004 | 0000,011D,0002,3AE8 |
| F009,E000,2780,0004 | 0000,011E,0002,3CF0 |
| F009,F000,27C0,0004 | 0000,011F,0002,3EF8 |

## Section 51, Subsection 0

This subsection first initializes Page Table and then does a random read/write/allocate.

Set Pattern:

Refer to section test patterns.

forands Used:

| Number | Description                     |
|--------|---------------------------------|
| 82     | Clear Stream and Purge All map. |
| 41     | Write CM 8 words).              |
| 40     | Read CM 8 words).               |
| 3F     | Read CM 32 words).              |

0000285 A

LMT3/P - 574

## Initialization:

- Clear Stream and Purge All maps.
- Start processor at micrand 82.
- Set PTA = 0.
- Set PTL = FF.

## Conditions:

- C.0
- Disable all map sets.
  - Write PTA in RF18 - R1F.
  - Write 8 descriptors in RF38 - RF3F (start with Page Table 1).
  - Start processor at micrand 41, write 8 descriptors in CM.
  - Enable map set x. (See note for map set selection.)
  - Write SVA in RF18 - RF1F.
  - Start processor at micrand 40, allocate 8 descriptors in page map x.
- If all 3 map sets are allocated, go to C.1; else go to C.0. (See note for map set selection.)
- C.1
- Write CM 100(HEX) - CM 11F(HEX) with patterns.
  - Enable all map sets.
  - Write SVA in RF0 - RF1F.
  - Clear RF20 - RF3F.
  - Start processor at micrand 3F, read 32 CM words.
  - Read and verify RF20 - RF3F, error if not equal data written in CM 100(HEX) - CM 11F(HEX).
  - Report data from last RF address read.
- C.2
- Report last RF address read in C.1.
- C.3
- First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit if error or all 3 Page Tables are done, else go to C.0.

Exit this section and skip to Section 51, Subsection 2 if PARAM14 is zero (default value).

00000285 B

LMT3/P - 575

# NOTE

Map sets are selected as follows:

| Page Table | Map Set |   |   |
|------------|---------|---|---|
| 1          | 0       | 1 | 2 |
| 2          | 2       | 0 | 1 |
| 3          | 1       | 2 | 0 |

## Section 51, Subsection 1

This subsection is a continuation of the previous subsection. All 3 Page Tables should have been allocated.

The subsection checks the final state of Page Table and Central Memory, and also checks the final state of page map by destroying Page Table and reading one word from each expected page descriptor in page map.

Test Pattern:

Refer to section.

Micrands Used:

| Number | Description              |
|--------|--------------------------|
| 42     | Write CM (32 locations). |
| 3F     | Read CM (32 locations).  |

Initialization:

- Refer to subsection 0.

Conditions:

- C.0
- If RI is set and iteration count is greater than zero, go to C.2.
  - Enable all map sets.
  - Write RF20 - RF3F with AAAA AAAA 5555 5555.
  - Write RF00 - RF1F with SVA in Page Table 4.
  - Start processor at micrand 42, write 32 words.
  - Read and verify CM 0 - CM 1F, error if not equal to data in Modified Page Table.
  - Report data from last CM address read.
- C.1
- Report last CM address read in C.0.

Conditions:

- C.2
- Clear CM 0 - CM 1F(HEX).
  - Write RF0 - RF1F with SVA (start with Page Table 1).
  - Start processor at micrand 3F, read CM (32 locations).
  - Read and verify RF20 - RF3F, error if not equal to AAAA AAAA 5555 5555(HEX).
  - Report data from last RF address read.
- C.3
- Report last RF address read in C.2.
- C.4
- First Failure Capture Information.
- Report PFS87 bits 19-23 in bits 59-63; information only.
  - Report last PTL written in bits 34-47; information only.
  - Report last PTA written in bits 13-31; information only.

Exit if error or all 3 Page Tables are used; else go to C.0.  
If RI is set, go to C.0.

# GLOSSARY

A

## Abs

Absolute.

## AC or A/C

Address Control.

## ACMIC

Address Control Functional Micrand.

## ADATB/RDSB/B

Refers to B Operand data in Multiply/ Divide unit test MDT3/P.

## ADATC/RDSC/C

Refers to C Operand data in Multiply/ Divide unit test MDT3/P.

## Adder

Refers to Large Adder in ALN test ANT3/P. Refer to Multiply Final Adder in test MDT3/P.

## Adder Bit 47 FF

Large Adder Result bit 47 flip-flop in ALN test ANT3/P.

## Adrs

Address.

## ALN

Arithmetic and Logical Network.

## ALU

Arithmetic and Logical Unit.

60000285 A

LMT3/P - A1

## AOR

Address-Out-of-Range.

## Arith

Arithmetic.

## ASCII

American Standard Code for Informa- tion Interchange.

## ASE

Address Specification Error.

## ASID

Active Segment Identifier.

## Assy

Assembly.

## Aux

Auxiliary.

## B

Refers to the operand input to ALN on the RSDB data path. Used in ALN test ANT3/P.

## BCD

Binary Coded Decimal.

## BD

Branch Delta.

## BDP

Business Data Processor.

60000285 A

LMT3/P - A2

Bfr

Buffer.

Bin

Binary.

Bkpt

Breakpoint.

IBN

Byte Number.

Bound

Boundary.

Br

Branch.

C

Refers to the operand input to the ALN on the RDSC data path. Used in ALN test ANT3/P.

CA

Condition Action.

CABR

Cache Address Buffer Register.

CAE

Condition Action Extension.

CAR

Cache Address Register.

80000285 A

LMT3/P - A3

CBP

Code Base Pointer.

CCR

Clock Condition Register.

CEL

Corrected Error Log.

C.GT.B

C Greater than B signal from the SKG in ALN test ANT3/P.

Chan

Channel.

Clk

Clock.

Clr

Clear.

CM

Central Memory.

CMC

Central Memory Control.

Cnt

Count.

Cntr

Counter.

80000285 A

LMT3/P - A4

Coef  
Coefficient.

Com  
Common.

Comp  
Complement.

Cond  
Condition.

Conf1  
Conflict.

Cont  
Control.

Conv  
Convert, Converter.

Cor  
Corrected.

CPU  
Central Processing Unit.

1CS  
Condition Sensing.

CSA  
Control Store Address.

60000285 A

LMT3/P - A5

CST  
Control Store.

CW1  
Control Word 1.

CW2  
Control Word 2.

DAI  
Data Interchange.

DCDR  
Decoder.

DCI  
Data control information. Two bytes of information that give the maintenance register address.

DEC  
Dependent Environment Control, Decimal.

Decr  
Decrement.

Descr  
Descriptor.

Destn  
Destination.

Disassy  
Disassembly.

60000285 A

LMT3/P - A6



Div

Divide.

Divisor Rgtr

Divide network register internal to LSI CI divide arrays.

DIyd

Delayed.

DMR

Debug Mask Register.

Dsb1

Disable.

DSS

Data Subfunction Select Micrand Field.

EBCDIC

Extended Binary Coded Decimal Inter- change Code.

ECC

Error Correction Code.

ECR

C170 Exit Mode Condition Register.

EI

Environment Interface.

EMMR or EMR

Exit Mode Mask Register.

60000285 A

LMT3/P - A7

Enb1

Enable.

Encdr

Encoder.

ESE

Environment Specification Error.

EXCH

Exchange.

Exp

Exponent.

Exponent Aux

Exponent Auxiliary board in ALN.

Ext

External.

Fctn

Function.

FF

Flip-flop.

FIFO

First In/First Out.

FLC

C170 Mode Central Memory Field Length.

60000285 A

LMT3/P - A8

FNO  
Fanout.

FRC  
Free Running Counter.

FU  
Functional Unit.

Gen  
Generator.

Gen1  
General.

GME  
Used by test CTT3/P to indicate a field (bits 62 and 63) within the general micrand.

Hldg  
Holding.

I  
Operand Register Designator.

IACY  
C170 Parcel Select Multiplexer (IFT3/P test).

IAHY  
Instruction Assembly Hybrid Multi- plexer (IFT3/P test).

IAPL  
C180 Parcel Select Multiplexer (IFT3/P test).

60000285 A

LMT3/P - A9

IB02  
Instruction Buffer Rank 02 (IFT3/P test).

IB12 valid  
Used in ALN test ANT3/P. Instruction Buffer Rank 12 valid signal from Instruction Fetch. This goes active when IF has finished fetching the next instruction from CM.

ICC  
Instruction Completion Control.

ICP  
Instruction Control Pipeline.

IDS  
Immediate Data Select Micrand Field.

IDX  
Indexed.

IF  
Instruction Fetch.

II  
Instruction Issue.

Immed  
Immediate.

Incr  
Increment.

Indef  
Indefinite.

60000285 A

LMT3/P - A10

Inf

Infinite.

Inh

Inhibit.

Inp

Input.

Instr

Instruction.

Int

Internal.

Integer Mux

Integer/Exponent Multiplexer in ALN.

Intrpt

Interrupt.

IOU

Input/Output Unit.

ISE

Instruction Specification Error.

IU

Instruction Unit.

J

Operand Register Designator.

80000285 A

LMT3/P - A11

JPS

Job Process State.

k

Operand Register Designator.

K

170 18-Bit Immediate Operand.

K/L

Key/Lock.

Kypt

Keypoint.

l

BDP Operand Length.

Ld

Load.

LM

Local Memory.

LMMIC

Local Memory Functional Micrand field.

LOS

Loss of Significance.

LRR

Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P, and ICT3/P to indicate a live register read.

80000285 A

LMT3/P - A12

LRU

Least Recently Used.

LRW

Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P and ICT3/P to indicate a live register write.

SD

Least Significant Digit.

LSI

Large Scale Integration.

LSM

Last Symbol Mask.

LSMPRE

LOAD/STORE Multiple Address Adder.

Lwr

Lower.

MAC

Maintenance Access Control.

Maint

Maintenance.

MAR

Micrand Address Register.

MBE

Multiple Bit Error.

60000285 A

LMT3/P - A13

MC

Master Clear.

MCH

Maintenance Channel.

MCMR

Monitor Condition Mask Register. Also refer to MMR.

MCR

Monitor Condition Register.

Mem

Memory.

Micr

Micrand.

Micrand Constant

Contents of ALN FU Micrand bits 0 through 6.

MIS

Miscellaneous Micrand Field.

Misc

Miscellaneous.

MMR

Monitor Mask Register.

MOP

Micro operation.

60000285 A

LMT3/P - A14

MPS

Monitor Process State.

MR

Maintenance Register.

MSB

Most Significant Bit.

MSC

Micrand Sequence Control. A field (bits 0 through 15) within the general micrand.

MSD

Most Significant Digit.

MSNZB

Most Significant Nonzero Byte.

Mult

Multiply.

Mux

Multiplexer.

Neg

Negative.

Norm

Normalize.

NPGK

New P Global Key.

60000285 A

LMT3/P - A15

NPLK

New P Local Key.

ns

Nanosecond.

o

BDP Operand Address.

Op

Operand.

Opcode

Operation Code.

OPGK

Old P Global Key.

OPLK

Old P Local Key.

OPI

Operand Issue.

Opn

Operation.

OSB

Operating System Bounds.

Out

Output.

60000285 A

LMT3/P - A16

Ovf1  
Overflow.

P  
Program Address.

Par  
Parity.

PDM  
Processor Detected Malfunction.

PE  
Parity Error.

PFA  
Page Frame Address.

PFS  
Processor Fault Status.

Ph  
Phase.

PIT  
Process Interval Timer.

PMF  
Performance Monitoring Facility.

PN  
Page Number.

60000285 A

LMT3/P - A17

PO  
Page Offset.

PONR  
Point Of No Return.

Pos  
Positive.

PP  
Peripheral Processor.

Prev  
Previous.

Pr1  
Priority.

Prod  
Product.

PSM  
Page Size Mask.

PSWF  
Page Table Search Without Find.

PTA  
Page Table Address.

PTD  
Page Table Descriptor.

60000285 A

LMT3/P - A18

PTL

Page Table Length.

PTM

Processor Test Mode.

PVA

Process Virtual Address.

PW

Partial Write.

Q

C180 16-Bit Immediate Operand.

Quad

Quadrant.

Quotient Rgtr

Divide network register internal to the LSI CI arrays.

R/W

Read/Write.

RAC

C170 Mode Central Memory Reference Address.

RAM

Random Access Memory.

RCL

Read and Clear Lock.

60000285 A

LMT3/P - A19

Rcvr

Receiver.

RDS

Register/Data Select.

RDSA

In ALN test ANT3/P, refers to the micrand field that provides the register file address on a write of ALN data and to the data path through DAI for that data.

RDSB

In ALN test ANT3/P, refers to the following data path to ALN. The specified register file is output to the RDSB bus, transferred to the ALN on ADATB, and is passed through the multiple/divide unit and output multiplexer to the general network.

RDSC

In ALN test ANT3/P, refers to following data path. The specified register file is sent to the RDSC bus, transferred to ADATC, then goes to the general network.

Ref

Reference.

Regen

Regenerator.

Rel

Relative.

Rem

Remainder.

Req

Request.

60000285 A

LMT3/P - A20

**Resync**

Resynchronize, Resynchronizer, Resynchronization.

**RF**

Register File.

**RFOx**

Used in ALN test ANT3/P and Multiply/ Divide test MDT3/P to identify the register file being used. x is 5, 6, 7, 8, 9, A, B, C, D, E or F (Hexadecimal).

**Rgtr**

Register.

**RMA**

Real Memory Address.

**ROM**

Read Only Memory.

**RSL**

Read and Set Lock.

**RS64**

Right Shift 64 signal from the SKG.

**Rxx**

Identifies ranks in IF and ICP in Test CTT3/P.

**SBE**

Single Bit Error.

**SC**

Soft Control.

60000285 A

LMT3/P - A21

**SCM**

Soft Control Memory.

**SCT**

Special Character Table.

**SECEDED**

Single Error Correction/Double Error Detection.

**Seg**

Segment.

**Sel**

Select.

**Sep**

Separate.

**Seq**

Sequence.

**Shift Mux**

Shift Network Multiplexer.

**Shifter**

Shift Network.

**Sig**

Significant.

**SIT**

System Interval Timer.

60000285 A

LMT3/P - A22



SKG

Shift Count Generator.

SM

Segment Map.

SMMIC

Segment Map Functional Micrand Field.

SMO(x)

Segment Descriptor Multiplexer Output bit x.

SMPDM

Segment Map Processor Detected Malfunction.

SN

Segment Number.

Spec

Specification.

SPID

Segment/Page Identifier.

STA

Segment Table Address.

STL

Segment Table Length.

Str

Stream.

60000285 A

LMT3/P - A23

Subtr

Subtract.

Suppr

Suppression.

SV

Specification Value.

SVA

System Virtual Address.

Sw

Switch.

Syn

Syndrome.

Sync

Synchronize.

Termn

Terminate.

TFA

Tag File Address.

ubit

Micrand Bit.

UCEL1

Uncorrected Error Log 1.

60000285 A

LMT3/P - A24

UCEL2

Uncorrected Error Log 2.

UCR

User Condition Register.

UMR

User Mask Register.

Unbr

Unbranch.

Uncond

Unconditional.

Uncor

Uncorrectable.

Undf1

Underflow.

Unlog

Unlogged.

Upr

Upper.

usec

Microsecond.

UTP

Untranslatable Pointer.

80000285 A

LMT3/P - A25

Ux

Used in Multiply/Divide test MDT3/P to identify Major Cycle Control bits 0 through 14.

VMID

Virtual Machine Identifier.

Vx

Used in Multiply/Divide test MDT3/P to identify Minor Cycle Control bits 0 through 15.

DS

Write Data Select.

WOI

Word of Interest.

Xfer

Transfer.

Xltr

Translator.

Xmtr

Transmitter.

XOR

Exclusive OR.

ZIF

Zero Insertion Force.

ZF

Zero Flag.

80000285 A

LMT3/P - A26

# LMT3/P MICRAND DEFINITIONS

## NOTE

The first word address of a particular micrand sequence in control store can be calculated by adding 200 hexadecimal to the micrand sequence number that is to be run, which is also a hexadecimal number.

THIS PAGE HAS INTENTIONALLY BEEN LEFT BLANK

| Number | Type                           | Characteristics                                                                                                                                                                                                                                                                                                          |
|--------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1      | Word Load (RMA)                | One 84-bit word is copied from CM address defined in Register File (RF) location 2) to RF location 8. RMA addressing is selected. Therefore, no address conversion is required.                                                                                                                                          |
| 2      | Stream Word Load (RMA)         | This micrand has the same characteristics as Micrand 1. The only difference is that Stream Mode is selected prior to execution and deselected after execution.                                                                                                                                                           |
| 3      | Word Store (RMA and Index)     | Bytes 2 through 7 (six bytes) are copied from RF location A to CM. The address (RMA) is defined in RF2.                                                                                                                                                                                                                  |
| 4      | Read Free                      | This micrand is similar to a Word Load, but because Running Counter of the function code, it reads the contents of the Central Memory Free Running Counter 17 times. The counts are stored in RF locations 8 through 16.                                                                                                 |
| 5      | Clear Residue Parity Errors    | This micrand issues a store word function to a null address. Nothing is actually stored but the address and data trunks in Address Control and Local Memory are cleared.                                                                                                                                                 |
| 6      | Test Resync Function           | This 4-micrand sequence uses the CMC Free Running Counter (FRC) to time the CMC Refresh cycle. The FRC is sampled only at Bank Zero time, so a start time from the FRC followed by a CMC Resync and an End Time from the FRC should show the Refresh cycle time. The first Resync is used to keep everything consistent. |
| 7      | Interrupt Processor (S00,SS03) | Interrupt the Processor. RDSc is set to 01 which selects CPUA. RMA mode (bit 35) is selected in the SMMIC field of the FU Micrand. The Interrupt status is sent to the processor Monitor Condition Register.                                                                                                             |

| Number | Type                               | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------|------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8      | Three Word Stores (RMA) (S00,SS04) | This micrand sequence copies three 64-bit words from RF locations A(HEX), B(HEX), and C(HEX) to CM addresses 00, 01, and 02 respectively. RMA addressing is selected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 9      | Prevalidate (RMA and TEST)         | This micrand is not expected to do anything except receive a response and exit.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| A      | Word Store (RMA)                   | One 64-bit word is copied from RF location A(HEX) to the CM address defined in RF location 2. RMA addressing is selected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| B      | Byte Store (RMA)                   | One to eight bytes are copied from RF location A (HEX) to the CM address defined in RF location 2. Byte length minus one (length -1) is in RF location 12(HEX). RMA addressing is selected.                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| C      | FAKECM Allocate Cache              | Allocate or assign one Cache Block (four words) to the address SVA) in RF location 2. FAKECM mode must be previously selected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| D      | Word Load (SVA)                    | One 64-bit word is copied from CM or Cache (address defined in RF location 2) to RF location 6. SVA addressing is selected. The PVA to SVA address conversion done in the Segment Map is bypassed. If FAKECM Mode is selected and the data requested is not in Cache, zeros are returned as data.                                                                                                                                                                                                                                                                                                                                                   |
| E      | Word Store (SVA)                   | One 64-bit word is copied from RF location A(HEX) to CM and Cache (if an address match occurs) at the address defined in RF location 2. SVA addressing is selected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| F      | Purge Cache                        | Clear Cache Tag Valid bits in all addresses of all available Cache sets.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 10     | Set P Descriptor                   | The P Descriptor Register holds the current ASID for all Instruction Fetch operations. This micrand initializes that register by using the Exchange Function. The following steps occur: <ol style="list-style-type: none"> <li>1. A Segment Map miss.</li> <li>2. This sends the STA Register plus the Segment Number from RF location 11(HEX) (both quantities are always zero) to CM.</li> <li>3. The new Segment Descriptor is read from CM 0. Its format is 0000 xxxx 0000 0000(HEX); where xxxx is the desired ASID.</li> <li>4. This quantity is stored in the P Descriptor Register and is used with all subsequent IF requests.</li> </ol> |

| Number | Type                      | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11     | Byte Store (SVA)          | One to eight bytes are copied from RF location A (HEX) to the CM address defined in RF location 2. Byte length minus one (length -1) is in RF location 12(HEX). SVA addressing is selected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 12     | Purge Page Map            | Clear the Page Map Valid bits in all locations of all Map Sets.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 13     | Set P Register (Start IF) | When the P Register is entered with a new byte address, Instruction Fetch (IF) sends the address to the Segment Map to pick up the current ASID and then to LM to fetch the data from the SVA (after SVA to RMA conversion). The following step chart explains the sequence. <ol style="list-style-type: none"> <li>1. The desired address PVA with Seg. No. = 0) is set in RF location 10(HEX).</li> <li>2. The micrand moves RF location 10(HEX) to the P Register.</li> <li>3. The P Register value is merged with the P Descriptor in the Segment Map to form the SVA.</li> <li>4. The IF Request and address are received at LM, the word is fetched from either Cache or CM depending on the circumstances and sent to the C180 Decode RAM.</li> <li>5. 100(HEX) is added to byte 0 of the data. This quantity is sent to Control Store as a CS address.</li> <li>6. The micrand at CS address 1xx(HEX) (xx = byte 0) is executed. This normally results in a constant being sent to RF location 14(HEX).</li> </ol> |
| 14     | Index                     | The Index operation is selected by setting the IIIDX bit in the LMMIC field of the FU micrand. The SVA address is in the RF location 2. The Index data is moved from LM to RF location 8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 15     | Test                      | The Test operation is selected by setting the IITST bit in the LMMIC field of the FU micrand. The SVA address is in RF location 2. The Test data is moved from LM to RF location 8.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 16     | Prevalidate (S23,S02)     | Same as Micrand E (Write Word to SVA), except the Valid bit in the ACMIC field is set in the FU micrand.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

| Number | Type                                   | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------|----------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17     | Word Store<br>(Multiple)<br>(S31,SS00) | <p>This micrand sequence is designed to time a Word Store operation. The first part reads the CMC Free Running Counter (FRC) (value saved in RF17) to establish a starting time.</p> <p>The second part copies a 64-bit data word from RF location A(HEX) to the CM address defined in RF location 2. The operation is repeated from one to 16 times. The repeat count (1 to F(HEX) is right-justified in RF location 13(HEX).</p> <p>The last part again reads the FRC (value saved in RF18) to record the ending time.</p> |
| 18     | Index<br>(Multiple)<br>(S31,SS00)      | <p>This micrand sequence is also bracketed with reads of the FRC, see micrand 17, to record the starting and ending times of the Index operation loop. The actual test executes an Index operation 1 to F(HEX) times. The SVA is in RF location 2, the data coming from LM is stored in RF location 6, and the repeat count, minus one (count -1) is in RF location 13(HEX).</p>                                                                                                                                             |
| 19     | Byte Load<br>(SVA)                     | <p>Copy one to eight bytes from CM or Cache (address defined in RF location 2) to RF location 6. Byte length minus one (length -1) in RF location 12 (HEX). I1DR is selected sending response to Address Control A-Stream. SVA addressing is selected.</p>                                                                                                                                                                                                                                                                   |
| 1A     | Double Word Load (I1/I1)               | <p>This micrand sequence consists of two micrands. Both perform word load operation; both send Instruction Issue (I1) requests only. The first micrand copies a word from CM or Cache (address defined in RF location 2) to RF location 6. The second micrand copies a word from CM or Cache (address defined in RF location 3) to RF location 7. Both have SVA addressing selected.</p>                                                                                                                                     |
| 1B     | Double Load (IF/SA)<br>(S35,SS00)      | <p>This micrand sequence starts Instruction Fetch (IF) at the PVA defined in RF location 10(HEX). This is followed by a Byte Load operation which is intended to cross a word boundary, causing a request to be issued from Address Control Stream A. One to eight bytes are copied from CM or Cache (address defined in RF location 2) to RF location 6. Byte length minus one (length -1) is in RF location 12(HEX). SVA addressing is selected.</p>                                                                       |

| Number         | Type                                                               | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
|----------------|--------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-----------|----------------------|---|------|------|---|-----|-----|---|-----|------|---|-----|-----|
| 1C             | Double Load (I1/IF)<br>(S35,SS00)<br>(S41,SS01)                    | <p>This micrand sequence first copies a word from CM or Cache (SVA defined in RF location 2) to RF location 6. This is followed by a Set P Register from RF location 10(HEX), which in turn starts IF at the PVA defined in RF location 10(HEX).</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 1D             | Double Word Load (I1/I1)<br>(Multiple)<br>(S35,SS01)<br>(S35,SS02) | <p>This micrand sequence consists of two double word load sequences. Both sequences are the same as micrand sequence 1A except for Register File locations.</p> <table> <tr> <th>Micrand Number</th><th>SVA is in</th><th>Destination Register</th></tr> <tr> <td>1</td><td>RF2</td><td>RF6</td></tr> <tr> <td>2</td><td>RF3</td><td>RF7</td></tr> <tr> <td>3</td><td>RF4</td><td>RF8</td></tr> <tr> <td>4</td><td>RF5</td><td>RF9</td></tr> </table> <p>This sequence may be looped on or repeated from 1 to 16 times. The loop count (1c) is defined in RF location 13(HEX) as 1c -1.</p> <p>Sequence timing is taken by reading the CMC Free Running Counter before (RF17) and after (RF18) the sequence loop.</p>    | Micrand Number | SVA is in | Destination Register | 1 | RF2  | RF6  | 2 | RF3 | RF7 | 3 | RF4 | RF8  | 4 | RF5 | RF9 |
| Micrand Number | SVA is in                                                          | Destination Register                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 1              | RF2                                                                | RF6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 2              | RF3                                                                | RF7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 3              | RF4                                                                | RF8                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 4              | RF5                                                                | RF9                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 1E             | Double Load (IF/SA)<br>(Multiple)<br>(S35,SS01)                    | <p>This micrand sequence consists of two double word load sequences. Both sequences are the same as micrand sequence 1B except for Register File locations.</p> <table> <tr> <th>Micrand Number</th><th>SVA is in</th><th>Destination Register</th></tr> <tr> <td>1</td><td>RF10</td><td>RF14</td></tr> <tr> <td>2</td><td>RF2</td><td>RF6</td></tr> <tr> <td>3</td><td>RFF</td><td>RF14</td></tr> <tr> <td>4</td><td>RF3</td><td>RF7</td></tr> </table> <p>This sequence may be looped on or repeated from 1 to 16 times. The loop count (1c) is defined in RF location 13(HEX) as 1c -1.</p> <p>Sequence timing is taken by Reading the CMC Free Running Counter before (RF17) and after (RF18) the sequence loop.</p> | Micrand Number | SVA is in | Destination Register | 1 | RF10 | RF14 | 2 | RF2 | RF6 | 3 | RFF | RF14 | 4 | RF3 | RF7 |
| Micrand Number | SVA is in                                                          | Destination Register                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 1              | RF10                                                               | RF14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 2              | RF2                                                                | RF6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 3              | RFF                                                                | RF14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |
| 4              | RF3                                                                | RF7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |      |      |   |     |     |   |     |      |   |     |     |

| Number         | Type                                                       | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
|----------------|------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|-----------|----------------------|---|-----|-----|---|------|------|---|-----|-----|---|-----|------|
| 1F             | Double Load<br>(II/IF)<br>(Multiple)<br>(S35,SS02)         | <p>This micrand sequence consists of two double word load sequences. Both sequences are the same as micrand sequence 1C except for Register File locations.</p> <table> <tr> <th>Micrand Number</th><th>SVA is in</th><th>Destination Register</th></tr> <tr> <td>1</td><td>RF2</td><td>RF6</td></tr> <tr> <td>2</td><td>RF10</td><td>RF14</td></tr> <tr> <td>3</td><td>RF3</td><td>RF7</td></tr> <tr> <td>4</td><td>RFF</td><td>RF14</td></tr> </table> <p>This sequence may be looped on or repeated from 1 to 16 times. The loop count (1c) is defined in RF location 13(HEX) as 1c -1.</p> <p>Sequence timing is taken by Reading the CMC Free Running Counter before (RF17) and after (RM18) the sequence loop.</p> | Micrand Number | SVA is in | Destination Register | 1 | RF2 | RF6 | 2 | RF10 | RF14 | 3 | RF3 | RF7 | 4 | RFF | RF14 |
| Micrand Number | SVA is in                                                  | Destination Register                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 1              | RF2                                                        | RF6                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 2              | RF10                                                       | RF14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 3              | RF3                                                        | RF7                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 4              | RFF                                                        | RF14                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 20             | Test and Set<br><br>Bit at SVA<br>(S36,SS02)<br>(S37,SS00) | <p>This micrand sequence will set a single bit in a CM byte.</p> <p>Also the value of the bit in CM prior to its being set is transferred to a Register File location bit 63.</p> <p>RF2 = The SVA byte address.<br/>RF6 = Where the old bit from CM is saved.<br/>RFA = Bit number within byte to set.</p>                                                                                                                                                                                                                                                                                                                                                                                                              |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 21             | Word Load<br>(SVA), Suppress<br>PSWF.<br>(S38,SS01)        | This micrand is the same as Micrand D (read word from SVA), except the IISUP bit is selected in the LMMIC field. This inhibits any Page Search Without Find status being generated by LM.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 22             | Purge/Bypass<br>Cache (SVA)<br>(S37,SS00)                  | Purge four Cache words (one block) at the SVA defined in RF location 2. Bit IIPCH is selected in the LMMIC field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 23             | Word Load<br>(SVA) Cache<br>Bypass                         | This micrand is the same as Micrand D (read word from SVA), only the IIBYP (Bypass Cache) bit is selected in the SMMIC field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |
| 24             | Word Store<br>(SVA) Cache<br>Bypass                        | This micrand is the same as Micrand E (write word to SVA), except the IIBYP (Bypass Cache) bit is selected in the SMMIC field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                |           |                      |   |     |     |   |      |      |   |     |     |   |     |      |

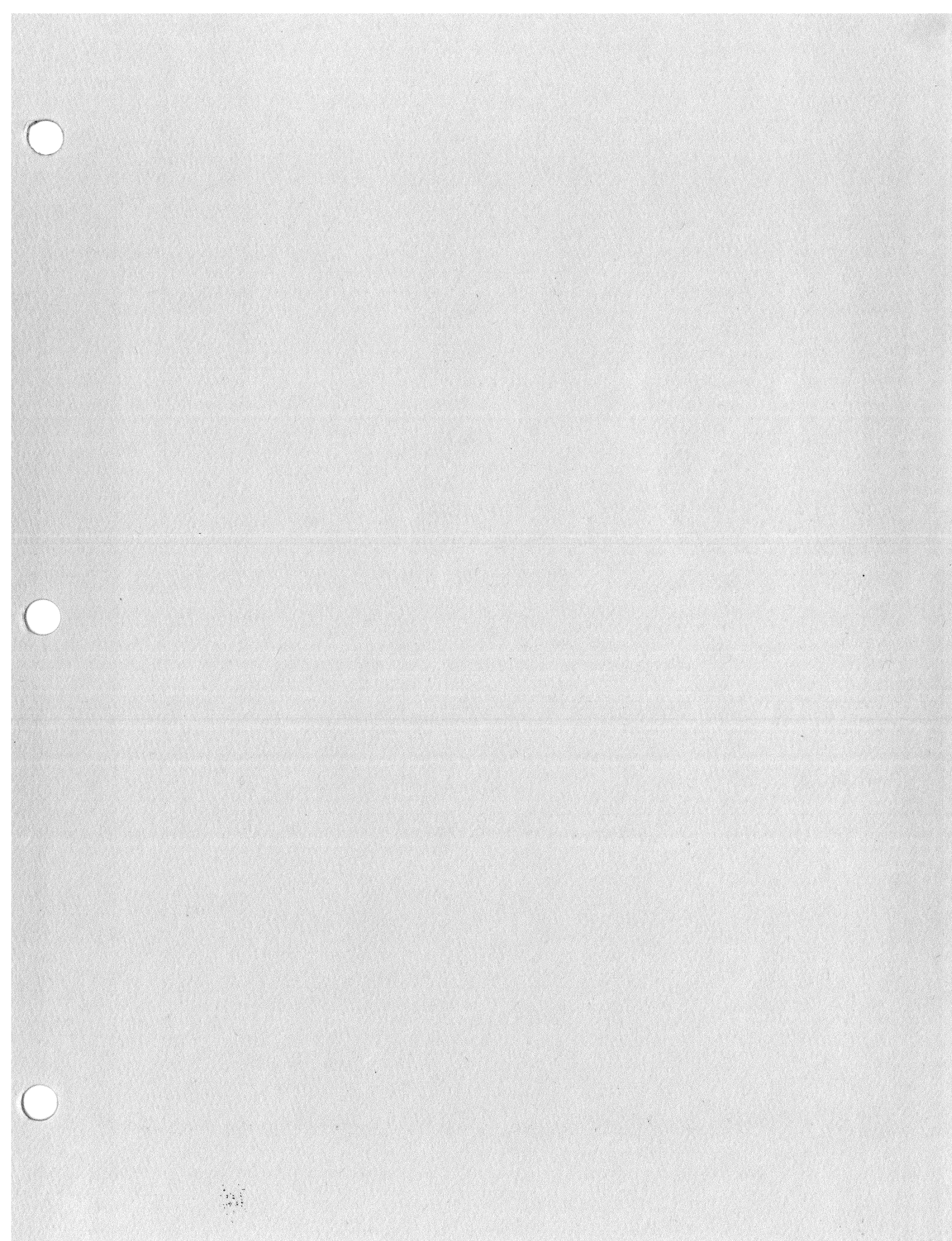
| Number | Type                                                   | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|--------|--------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 25     | Sense NMRO<br>Status<br>(S41,SS02)                     | This micrand only senses the No Memory Requests Outstanding status. This is sensed by setting the CS field of the MSC (General Micrand) to 1F(HEX). If NMRO is true a second micrand is executed that copies a constant (E7--E7) from the 64-bit FU micrand to RF location 6. If false (memory active) RF location 6 is left as is.                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 26     | Triple Word<br>Load (SVAs)<br>(II/IF/II)<br>(S41,SS02) | <p>This micrand sequence issues three requests to LM for data. Before the requests are sent a micrand is issued to resync the CM Refresh Counter. The first request is a Word Load and the request is issued from Instruction Issue (II). One word is copied from CM or Cache (address defined in RF location 2) to RF location 6. The second request comes from Instruction Fetch (IF) and results when the P Register is loaded from RF location 10(HEX). The data word fetched addresses a Control Store micrand which in turn sets RF location 14(HEX) with a 64-bit constant.</p> <p>The third request again is issued by II and is the same as the first, except the CM or Cache address is in RF location 3 and the destination register is RF location 7.</p> |
| 27     | Purge Cache<br>Write Word<br>(S11,SS03)                | This micrand sequence issues two micrands, the first is a Purge Cache All (same as Micrand F). The second is a Word Store using a SVA. The data in RFA is written to the SVA in RF2 (same as Micrand E).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 28     | Purge Cache<br>or Map<br>(S12<br>and S27)              | This micrand is normally written into Control Store at address 105(HEX) before issuing a Set P Register (Micrand 13) to start Instruction Fetch. A constant of 050x is expected at CM address zero, which, when fetched by IF will address micrand 28 at CS 105 (HEX). The x represents the purge code and is placed in the 4-bit k field which is passed on to LM from the virtual instruction. The Purge 1 bit is set in the SMMIC field. RF location 2 holds the purge address SVA).                                                                                                                                                                                                                                                                               |

| Number | Type                                                                                | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|--------|-------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29     | Word Store/<br>Word Load (SVA)<br>(11/11)<br>(Multiple)<br>(S48,SS00)<br>(S48,SS01) | This micrand sequence is designed to time a write/read operation. The first and last micrands read the CMC Free Running Counter (FRC) to determine the elapsed time of the test sequence. The starting time is read to RF location 17(HEX), the end time is read to RF location 18(HEX). The test sequence, or micrands between the FRC reads, consist of two micrands. The first is a word store as explained in Micrand E; the second is a word load as explained in Micrand D. The word store copies data from RF location A(HEX) to CM and Cache (if hit), (SVA defined in RF location 3). The word load copies data from CM or Cache (SVA defined in RF location 2) to RF location 6.<br><br>The test sequence (write/read operation) may be repeated from 1 to 18 times. The loop count (1c) is defined in RF location 13(HEX) as 1c-1. |
| 2C     | IF/Word<br>Load (SVA)<br>(S48,SS00)                                                 | This micrand sequence issues two requests to LM, both request 64-bit words. The first request comes from IF and copies one 64-bit word from the PVA in RF location 10 to the IF Unit. RF location 14(HEX) is written by the resulting micrand execution. The second request comes from II and copies one 64-bit word from CM SVA in RF location 2) to RF location 6.                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 2D     | IF/Delay/<br>Word Load (SVA)<br>(S48,SS00)                                          | This micrand sequence is the same as sequence 2C except a series of Inhibit - FU (No-op) micrands are placed between the IF and the Word Load to provide the delay.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 2E     | IF/Delay/<br>Word Load (SVA)<br>(S48,SS01)                                          | Same as sequence 2D except the length of the delay is adjusted to meet S48,SS01 requirements.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 2F     | IF/Word<br>Store (SVA)<br>(S48,SS01)                                                | This micrand sequence issues two requests to LM, the first from IF fetches one 64 bit word from CM PVA in RF location 10(HEX) to the IF Unit. RF location 14 is written by a subsequent micrand. The second request comes from II and copies one 64-bit word from RF location A(HEX) to CM SVA in RF location 2).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 30     | IF/Delay/<br>Word Store<br>(SVA)<br>(S48,SS01)                                      | This sequence is the same as sequence 2F except a delay is inserted between the IF and II requests. Inhibit - FU micrands comprise the delay.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| Number | Type                                         | Characteristics                                                                                                                                                                                                                                                                                                                                                                  |
|--------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 31     | Stream Word<br>Load (RMA to<br>RF0-RFF)      | Copy a variable number (1 through 18) of words from CM or Cache starting at the address (RMA) defined in RF location 12(HEX) to a sequential set of Register File locations between RF location 0 and F(HEX). The number of words to copy minus one (words -1) is in RF location 13(HEX). Stream Mode is selected.                                                               |
| 32     | Stream Word<br>Load (RMA to<br>RF10-RF1F)    | Copy a variable number (1 through 18) words from CM or Cache starting at the address (RMA) defined in RF location 2 to a sequential set of Register File locations between RF location 10(HEX) and RF location 1F(HEX). The number of words copied is in RF location 3(HEX). The number is one less than the words copied; i.e., 0 would copy one word. Stream Mode is selected. |
| 33     | Stream Word<br>Load (RMA to<br>RF20-RF2F)    | Same as Micrand 32, except the Register File locations copied to are between RF location 20(HEX) and RF location 2F(HEX).                                                                                                                                                                                                                                                        |
| 34     | Stream Word<br>Load (RMA to<br>RF30-RF3F)    | Same as Micrand 32, except the Register File locations copied to are between RF location 30(HEX) and 3F(HEX).                                                                                                                                                                                                                                                                    |
| 35     | Stream Word<br>Store (RMA from<br>RF0-RFF)   | Copy a variable number (1 through 18) of words from a sequential group of Register File locations between RF locations 0 and F(HEX) to CM and Cache (if hit) starting at the RMA defined in RF location 12(HEX). The number of copies is in RF location 13(HEX). The number is always one less than the number of words copied. Stream Mode is selected.                         |
| 36     | Stream Word<br>Store (RMA from<br>RF10-RF1F) | Same as Micrand 35, except the Register File locations copied from are RF locations 10(HEX) and 1F(HEX). Also the CM starting address is in RF location 2, and the word count is in RF location 3.                                                                                                                                                                               |
| 37     | Stream Word<br>Store (RMA from<br>RF20-RF2F) | Same as Micrand 36, except the Register File locations copied from are RF location 20(HEX) and 2F(HEX).                                                                                                                                                                                                                                                                          |
| 38     | Stream Word<br>Store (RMA from<br>RF30-RF3F) | Same as Micrand 36, except the Register File locations copied from are RF locations 30(HEX) and 3F(HEX).                                                                                                                                                                                                                                                                         |

| Number | Type                                      | Characteristics                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|--------|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 39     | Move A5A5<br>Constant to<br>Register File | This micrand normally occupies the 100(HEX)-word block in Control Store from address 100(HEX) through 1FF(HEX). The operation simply copies its FU portion to Register File location 14(HEX). The data in the FU portion is formatted as A5A5 A5A5 A5A5 xyxy(HEX); xy = the Control Store address of this particular micrand minus 100(HEX). This micrand is normally executed as a result of an Instruction Fetch operation. The result stored in RF14 indicates what Control Store address was accessed. |
| 3A     | Sense NMRO<br>Status<br>(S41,SS02)        | This micrand is the same as micrand sequence number 1A, except after execution of the double word load it jumps to Micrand 25 to sense Memory Requests Outstanding instead of going to halt.                                                                                                                                                                                                                                                                                                               |
| 3B     | Test CNTLPE<br>(S45,SS01)                 | This micrand is used during the testing of the CNTLPE Parity Error Signals. See Section 45, Subsection 1 description for additional detail.                                                                                                                                                                                                                                                                                                                                                                |
| 3C     | Loop CPU<br>(S39,SS02)                    | This micrand will put the CPU in a loop doing nothing. There is no exit. It is used during the Cache Invalidate Address Parity Check. See Section 39.2, Subsection 2 for details.                                                                                                                                                                                                                                                                                                                          |
| 3D     | IF/Word<br>Load<br>(S47,SS00)             | This micrand sequence is used to create a Wait Allocate conflict. The second IF request must wait for the II Word Load to finish its block fill before it can be allocated.                                                                                                                                                                                                                                                                                                                                |
| A0     | No-op Delay<br>String<br>(S49,SSxx)       | This micrand sequence consists entirely of Inhibit - FU (No-op) micrands. The number depends on the length of time it takes from the Start Processor at the top of the string until an Invalidate Cache request reaches LM. The last Inhibit - FU micrand in the sequence is always modified to jump to another micrand or micrand sequence.                                                                                                                                                               |









4201 North Lexington Avenue Saint Paul, Minnesota 55126-6198